EntDetector: entanglement detecting toolbox for bipartite quantum states

Roman Gielerak[1][0000−0001−8657−0829], Marek Sawerwain[1][0000−0001−8468−2456], Joanna Wiśniewska[2][0000−0002−2119−3329], and Marek Wróblewski[1][0000−0002−7249−8579]

[1] Institute of Control & Computation Engineering
University of Zielona Góra, Licealna 9, Zielona Góra 65-417, Poland
{R.Gielerak, M.Sawerwain, M.Wróblewski}@issi.uz.zgora.pl
[2] Institute of Information Systems, Faculty of Cybernetics,
Military University of Technology, Gen. S. Kaliskiego 2, 00-908 Warsaw, Poland
JWisniewska@wat.edu.pl

Abstract. Quantum entanglement is an extremely important phenomenon in the field of quantum computing. It is the basis of many communication protocols, cryptography and other quantum algorithms. On the other hand, however, it is still an unresolved problem, especially in the area of entanglement detection methods. In this article, we present a computational toolbox which offers a set of currently known methods for detecting entanglement, as well as proposals for new tools operating on two-partite quantum systems. We propose to use the concept of combined Schmidt and spectral decomposition as well as the concept of Gramian operators to examine a structure of analysed quantum states. The presented here computational toolbox was implemented by the use of Python language. Due to popularity of Python language, and its ease of use, a proposed set of methods can be directly utilised with other packages devoted to quantum computing simulations. Our toolbox can also be easily extended.

Keywords: quantum entanglement · quantum software · numerical computations.

1 Introduction

Quantum entanglement [20], [2], [15] is the physical phenomenon, already described in [5] by Einstein, Podolsky, Rosen. Currently, quantum entanglement for quantum states [18] is the basis of many quantum information processing protocols such as teleportation [4], communication [10] and cryptographic protocols as well [3].

On the other hand, the quantum entanglement phenomenon is still not a well-understood problem. One of the main issues is the criterion for detecting entanglement [7] in quantum pure and mixed states. In the case of bipartite pure states, the Schmidt criterion can be successfully applied – it gives an unambiguous answer whether we are dealing with an entangled state. However, in the

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

case of quantum states described by a density matrix, so-called mixed states, the problem of detecting entanglement is still not solved in computationally effective way. The general criterion of entanglement detection for density matrix cases is related to the entanglement witness theory [14], and so far, there is no simple and fast (especially computationally) entanglement verification criterion for such type of states. In fact due to the proven NP-hardness [12], [6] of the problem "entangled or separable?", it is hardly to expect that such efficient algorithms do exist on the classical side of computational technologies.

In this paper, we present a selected set of computational methods devoted to numerical studies of bipartite entanglement in quantum states. The aim is to develop a publicly available set of functions in Python, which will allow utilising the proposed package of computational methods also in combination with other packages related to quantum computing, such as QuTiP [16] and Qiskit [1].

In addition to the implementation of the basic generally known methods, our EntDetector package also offers an access to methods of entanglement testing using so-called Gramian matrices [8], as an additional tool for checking the entanglement level between individual qubits in a given pure bipartite state. The Gramian calculation techniques also allow us to provide a new method of easy calculation of the density matrix in the case of a bipartite system.

We use Python language in our package, because it is currently very popular in the field of quantum computing simulation. It seems that available Python software do not offers enough tools in area of entanglement detection although, of course, it is necessary to indicate the existence of the packages like Qubit4Matlab [23] or QETLAB [17], which offer a support in this field. However, they require a Matlab software [19].

The article is organized as follows: section 2 introduces the basic concepts and briefly describes the basic math engine that is used in the EntDetector package. The technical aspects of the package, including examples of usage, are presented in section 3. A summary is provided in section 4. This paper ends with acknowledgements and bibliography sections.

2 Mathematical framework

2.1 Spectral and Schmidt decomposition

Let $\mathcal{H} = \mathbb{C}^{d^2} = \mathbb{C}^d \otimes \mathbb{C}^d$ be a bipartite (A and B) finite dimensional Hilbert space and let $Q \in E(\mathcal{H})$ be a given quantum state. It is well known that the spectrum of Q (counting multiplicity) $\sigma(Q) = (\lambda_1, \lambda_2, \ldots, \lambda_{d^2})$, is purely discrete (it is important to point out that the list forming spectrum is ordered in non-increasing way) and the following spectral decomposition is valid:

$$Q = \sum_{i=1}^{d^2} \lambda_i |\Psi_i\rangle\langle\Psi_i| \tag{1}$$

where the orthogonal (and normalised) system of eigenfunctions $|\Psi_i\rangle$ of Q forms a complete orthonormal system of $\dim \mathcal{H} = d^2$.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

It is important to point out in this moment that the material presented in this report is valid in the current form only under the assumption that the spectrum of Q is simple which means that all of the corresponding eigenvalues are non-degenerated. The general case is technically more involved and will be presented (due to the limited space here) in an another paper [9].

Each eigenfunction $|\Psi_i\rangle$ can be expanded by the use of the Schmidt decomposition [11]:

$$|\Psi_i\rangle = \sum_{j=1}^{d} \tau_j^i |\psi_j^i\rangle \otimes |\vartheta_j^i\rangle, \tag{2}$$

where $\tau_j^i \geq 0$, $\sum_{j=1}^{d} (\tau_j^i)^2 = 1$ and systems $\{\psi_j^i\}$ form a complete orthonormal systems in part A, and resp. $\{\vartheta_j^i\}$ in B.

Let us define the following matrix SaSD(Q) (named Schmidt and Spectral Data) connected to the analysed state Q: it is $(d^2, (d+1))$ matrix in which we collect all the appearing Schmidt coefficients τ_i^n in the (d^2, d) block building from the first d^2 rows and d columns of SaSD(Q), and in the last column we localize the eigenvalues of Q. Graphically the map SaSD defined on $E(\mathcal{H})$ looks like:

$$\mathrm{SaSD}(Q) = \left(\begin{array}{ccc|c} \tau_1^1 & \dots & \tau_d^1 & \lambda_1 \\ \vdots & \ddots & \vdots & \vdots \\ \tau_1^{d^2} & \dots & \tau_d^{d^2} & \lambda_{d^2} \end{array} \right), \tag{3}$$

or in coordinates:

$$\begin{aligned} \mathrm{SaSD}(Q)_{j,i} &= \tau_j^i \quad \text{for } j = 1 : d^2, i = 1 : d, \text{ and} \\ \mathrm{SaSD}(Q)_{j,d+1} &= \lambda_j \quad \text{for } j = 1 : d^2. \end{aligned}$$

The map SaSD is uniquely defined (all eigenvalues are different) for each Q. However:

Proposition 1. *Let U_1, U_2 be a pair of unitary maps in $\mathbb{C}^d$ i.e. U are $SU(d)$ group elements, then we define its action on Q as*

$$Q \to (U_1^\dagger \otimes U_2^\dagger)(Q)(U_1 \otimes U_2), \tag{4}$$

then

$$\mathrm{SaSD}\left((U_1^\dagger \otimes U_2^\dagger)(Q)(U_1 \otimes U_2)\right) = \mathrm{SaSD}(Q). \tag{5}$$

This means that on each $2(d^2-1)$ – dimensional orbit of the local action of the group $SU(d)$ in the space $E(\mathcal{H})$ of dimension (real) d^4–1, the $(d^2(d+1) - (d^2+1))$-dimensional table SaSD do not change its value. For the pair of qudits of an arbitrary dimensions from Eq. (5) it follows that the matrix SaSD is locally unitary invariant. However, still the complete knowledge of SaSD(Q) is not sufficient for the complete (modulo $SU(d) \otimes SU(d)$) determination of Q as there is still deficit in dimensions at least as large as $\delta = d^4 - d^3 - 2d^2 + 2$, which for the case of two qubits gives $\delta = 2$.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

Example 1. Let us assume that we have states Q_p, Q_s with the corresponding SaSD tables:

$$\mathrm{SaSD}(Q_p) = \left(\begin{array}{ccc|c} \tau_1^1 & \dots & \tau_n^1 & 1 \\ 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \dots & 0 & 0 \end{array}\right), \quad \mathrm{SaSD}(Q_s) = \left(\begin{array}{ccc|c} 1 & \dots & 0 & \lambda_1 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \dots & 0 & \lambda_{d^2} \end{array}\right). \tag{6}$$

Then Q_p is a pure state and Q_s is a separable state.

Remark 1. It should be noted, once again, that in this part of our paper we only discussed case when the spectrum of Q is simple. In general case, the situation is more complicated and will be discussed in an another paper [9].

Any scalar function defined on the basis of $\mathrm{SaSD}(Q)$ will be locally $SU(d) \otimes SU(d)$ invariant scalar. In particular, the function that is named von Neumann entropy (vNEN) of the SaSD, which is defined below:

$$\mathrm{vNEN}(Q) = \sum_{i=1}^{d^2} \sum_{j=1}^{d} \Big(- (\tau_j^i)^2 \log \Big((\tau_j^i)^2 \Big) \Big) - \sum_{i=1}^{d^2} \lambda_i \log(\lambda_i), \tag{7}$$

is monotone non-increasing under the action of any local quantum operations [9] and invariant under the action of local unitary operations, where the standard convention $0 \cdot \log 0 = 0$ is being used.

Proposition 2. *The function* vNEN*, defined on $E(\mathcal{H})$, is continuous in $||\cdot||_1$ topology and is $SU(d) \otimes SU(d)$ invariant.*

Remark 2. An extensive study of the introduced von Neumann entropy and other locally unitary invariant functions build on SaSD, and, what is even more important, the study of functions which are monotonic (in the sense of majorization theory) and also in the sense of (S)LOCC semi-order relation will be presented elsewhere [9].

Remark 3. Some computational examples of the use of SaSD are presented at point 3.3 of this paper.

From the very definition of vNEN it follows that

$$\sup_{Q \in E(\mathcal{H})} \mathrm{vNEN}(Q) = (d^2 + 2) \log d, \tag{8}$$

It easy to observe that the SaSD table on which the introduced global entropy vNEN attains its maximal value looks like :

$$\mathrm{SaSD}(Q) = \left(\begin{array}{ccc|c} \frac{1}{\sqrt{d}} & \cdots & \frac{1}{\sqrt{d}} & \frac{1}{d^2} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{1}{\sqrt{d}} & \cdots & \frac{1}{\sqrt{d}} & \frac{1}{d^2} \end{array}\right), \tag{9}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

and corresponds to the $(d^2-1)^2$ – dimensional manifold of maximally mixed states in $E(\mathcal{H})$.

The extended SaSD table, denoted as exSaSD, is obtained from SaSD by adding to SaSD, defined in Eq. 3, two additional columns obtained from Schmidt decompositions (Eq. 2):

$$\mathrm{exSaSD}(Q) = \left(\begin{array}{ccc|c|ccc|ccc} \tau_1^1 & \dots & \tau_d^1 & \lambda_1 & \psi_1^1 & \dots & \psi_d^1 & \vartheta_1^1 & \dots & \vartheta_d^1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \tau_1^{d^2} & \dots & \tau_d^{d^2} & \lambda_{d^2} & \psi_1^{d^2} & \dots & \psi_d^{d^2} & \vartheta_1^{d^2} & \dots & \vartheta_d^{d^2} \end{array}\right). \tag{10}$$

Locally available information on state Q, as is well known, is contained in the corresponding density matrices. Owing to the obtained below decomposition, the computation of the corresponding reduced density matrices is very easy now.

Taking Eq. 2 into account, and after few lines of calculations, we obtain:

$$Q^B = \mathrm{Tr_A}\left(Q\right) = \mathrm{Tr_A}\left(\sum_{i=1}^{d^2} \lambda_i |\Psi_i\rangle\langle\Psi_i|\right) = \sum_{i=1}^{d^2} \lambda_i Q_i^B, \tag{11}$$

where the operators $Q_i^B = \sum_{j=1}^{d} |\tau_j^i|^2 |\vartheta_j^i\rangle\langle\vartheta_j^i|$ are states on the subsystem B.

Similarly, for the reduced density matrix connected to the observer relating to the subsystem A:

$$Q^A = \mathrm{Tr_B}\left(Q\right) = \mathrm{Tr_B}\left(\sum_{n=1}^{d^2} \lambda_n |\Psi_n\rangle\langle\Psi_n|\right) = \sum_{n=1}^{d^2} \lambda_n Q_n^A, \tag{12}$$

where $Q_n^A = \sum_{i=1}^{d} |\tau_i^n|^2 |\psi_i^n\rangle\langle\psi_i^n|$ are states, this time, on the subsystem A.

The obtained systems of operators $\{Q_n^A\}$ and $\{Q_n^B\}$ are consisting of non-negative, and therefore Hermitian, operators and are locally measurable, each of them. In particular, the squares of the Schmidt coefficients τ_i^n in the Schmidt decompositions of the parent state Q eigenfunctions are observable (measurable) quantities.

Proposition 3. *Let $\mathcal{H} = \mathbb{C}^{d^2} = \mathbb{C}^d \otimes \mathbb{C}^d$ be a bipartite Hilbert space and let $Q \in E(\mathcal{H})$. Let (Q^A, Q^B) be the pair of corresponding reduced density matrices and let*

$$Q_n^A = \sum_{i=1}^{d} |\tau_i^n|^2 |\psi_i^n\rangle\langle\psi_i^n| \quad \text{and corr.} \quad Q_n^B = \sum_{i=1}^{d} |\tau_i^n|^2 |\vartheta_i^n\rangle\langle\vartheta_i^n|, \tag{13}$$

for $n = 1 : d^2$ be the corresponding operators as defined by the use of $\mathrm{exSaSD}(Q)$. *Then $Q^A = \sum_{n=1}^{d^2} \lambda_n Q_n^A$ and $Q^B = \sum_{n=1}^{d^2} \lambda_n Q_n^B$.*

As the local information is invariant under local unitary action, we can define, in according to the ideas presented in [8], the following interesting invariant and monotonic functions.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

Definition 1. *Gramian functions of Q:*

$$G(Q_n^A) = \det\left(1_{\mathbb{C}^d} + Q_n^A\right) = \prod_{j=1}^{d}\left(1 + (\tau_j^n)^2\right), \tag{14}$$

for $n = 1{:}d^2$ and $1_{\mathbb{C}^d}$ represents an identity operator in space of the subsystem A. Identical definitions are also valid in part B of the analysed system.

Proposition 4. *For each n the value $G(Q_n^A)$ is invariant under the action of local unitary group, for any unitary map U acting on $\mathbb{C}^d$:*

$$G(UQ_n^AU^\dagger) = G(Q_n^A) \quad \text{and resp.} \quad G(UQ_n^BU^\dagger) = G(Q_n^B). \tag{15}$$

Proof. Obvious. □

Lemma 1. *For each $n = 1{:}d^2$,*

$$\sup_{Q\in E(\mathbb{C}^{d^2})} G(Q_n^A) = \sup_{Q\in E(\mathbb{C}^{d^2})} G(Q_n^B) = \prod_{j=1}^{d}\left(1 + \frac{1}{d}\right), \tag{16}$$

is true.

Let us define also logarithmic volumes of Q_n^A (resp. Q_n^B) as:

$$g^A(n) = \log G(Q_n^A) = \log G(Q_n^B) = g^B(n), \tag{17}$$

and also the complete Gram volumes of Q as

$$\mathfrak{g}^A(Q) = \prod_{k=1}^{d^2} G(Q_k^A) = \mathfrak{g}^B(Q) = \prod_{k=1}^{d^2} G(Q_k^B). \tag{18}$$

Together with its logarithmic counterparts :

$$\mathfrak{lg}^A(Q) = \log\prod_{k=1}^{d^2} G(Q_k^A) = \mathfrak{lg}^B(Q) = \log\prod_{k=1}^{d^2} G(Q_k^B). \tag{19}$$

Proposition 5. *The complete Gram volume $\mathfrak{g}^A(Q)$ of a given $Q \in E(\mathbb{C}^{d^2})$ and its logarithmic counterpart are locally $SU(d)\otimes SU(d)$ invariants of Q. The same is true for part B of the analysed system. Additionally:*

$$\sup_{Q\in E(\mathbb{C}^{d^2})} \mathfrak{g}^A(Q) = \sup_{Q\in E(\mathbb{C}^{d^2})} \mathfrak{g}^B(Q) = \left(\prod_{j=1}^{d}\left(1 + \frac{1}{d}\right)\right)^{d^2}, \tag{20}$$

and

$$\sup_{Q\in E(\mathbb{C}^{d^2})} \mathfrak{lg}^A(Q) = \sup_{Q\in E(\mathbb{C}^{d^2})} \mathfrak{lg}^B(Q) = d^2\left(\sum_{j=1}^{d}\log\left(1 + \frac{1}{d}\right)\right). \tag{21}$$

Another approach to certain aspects of reduced density matrices structure is based on the use of the Schmidt decomposition method in the Hilbert-Schmidt space of operators build on the space $\mathcal{C}^d\otimes\mathcal{C}^d$. This will be discussed in a separate paper [9].

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

2.2 Some remarks on the majorization theory

For a given finite sequence $\underline{a} = (a_1, \ldots, a_n)$ where $a_i \in \mathbb{R}$, we apply the operation of ordering in non-increasing order and denote the result as $\underline{a^{\geq}}$. Of particular interest, is the image of this operation when applied point-wise to the finite dimensional simplex $C_+^n(1) := \{\underline{a} = (a_1, \ldots, a_n), a_i \in \mathbb{R}, a_i \geq 0, \sum_{i=1}^n a_i = 1\}$. This will be denoted as $C^{\geq}$.

Let us recall the standard definition of majorizations. Let $a, b \in C^{\geq}$. Then we say that $\underline{b}$ majorizes $\underline{a}$ iff:

$$\sum_{i=1}^{k} a_i \leq \sum_{i=1}^{k} b_i, \tag{22}$$

for any $k = 1 : n$.

If this holds to be true then we denote this fact as $\underline{a} \preceq \underline{b}$.

We say that $\underline{b}$ majorizes multiplicatively $\underline{a}$ iff for any k, and $k = 1 : n$:

$$\prod_{i=1}^{k} (a_i + 1) \leq \prod_{i=1}^{k} (b_i + 1). \tag{23}$$

If this holds to be true then we denote this fact as $\underline{a} \preceq_m \underline{b}$.

Let F be any function (continuous, but not necessarily) on the interval $[0, 1]$. Let us recall the well known result, see i.e. [2].

Lemma 2. *Let as assume that f is a continuous, increasing and convex function on $\mathbb{R}$. If $\underline{a} \preceq \underline{b}$ then $f(\underline{a}) \preceq f(\underline{b})$.*

It is clear from the very definition that $\underline{a} \preceq_m \underline{b}$ iff $\log(\underline{a} + 1) \preceq \log(1 + \underline{b})$.

Proposition 6. *Let $\underline{a}, \underline{b} \in C^{\geq}$ and let us assume $\underline{a} \preceq_m \underline{b}$. Let f be continuous, increasing function such that the composition $f \circ \exp(x)$ is convex on a suitable domain. Then $f(\underline{a}) \preceq f(\underline{b})$.*

Proof. Fixing $k, k = 1 : n$, the following holds to be true:

$$\prod_{i=1}^{k} (a_i + 1) \leq \prod_{i=1}^{k} (b_i + 1). \tag{24}$$

Taking log of both side we obtain

$$\sum_{i=1}^{k} \log(1 + a_i) \leq \sum_{i=1}^{k} \log(1 + b_i). \tag{25}$$

Applying Lemma 2, we get

$$\sum_{i=1}^{k} f(e \cdot a_i) \leq \sum_{i=1}^{k} f(e \cdot b_i), \tag{26}$$

and the symbol e represents the Euler constant. □

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

In particular, taking constant function $f(x) = x$, we conclude:

Corollary 1. *Let $\underline{a}, \underline{b} \in C^{\geq}$ and let us assume that $\underline{a} \preceq_m \underline{b}$, then $\underline{a} \preceq \underline{b}$.*

The last result says that each linear chain of the semi-order relation $\preceq_m$ in $E(H)$ is contained in some linear chain of the semi-order. It means that the semi-order $\preceq_m$ is finer than those induced by $\prec$.

Theorem 1. *Let $\mathcal{H}$ be a separable Hilbert space and let Q_1 and Q_2 be states on $\mathcal{H}$. Then any $\preceq$-maximal element in $E(\mathcal{H})$ is also $\preceq_m$-maximal.*

Proof. If $\sigma(Q_1) \preceq_m \sigma(Q_2)$ then $\sigma(Q_1) \prec \sigma(Q_2)$. Let Q^* be a $\prec$-maximal in $E(H)$, and let as assume that there exists $Q\#$ such that $Q^* \preceq_m Q\#$ and the contradiction is present. □

3 Implementation and example in Python environment

In this part of the article, we present a basic information concerning the EntDetector package. The most important assumptions, about the implementation and realised functionalities, are described in section 3.1. The examples of our package's use are shown in section 3.2.

In section 3.3, we join the spectral and Schmidt decomposition for pure and mixed states. Our package carries out the decomposition which, as it was shown in section 2.1, allows characterising a bipartite quantum state by the level of entanglement e.g. for isotropic states, as shown in section 3.4.

3.1 Implementation assumptions

One of the fundamental assumptions about the EntDetector implementation was the choice of Python programming language because of its accessibility and simplicity. Another crucial issue is the prospect of utilizing other known libraries for quantum computation, like qiskit [1] and QuTiP [16], which are available in Python. We also assume that the realisation of numerical calculations are performed by the packages NumPy [13] and SciKit [24].

The mentioned simplicity of the EntDetector's usage may be shown on the example of an access to basic functions generating quantum states. We have prepared representations of pure states (e.g. Bell, GHZ, W states) and density matrices (e.g. Horodecki (2×4) and (3×3) bound entangled states, isotropic states) which are invoked in the following way:

- `q = create_qubit_bell_state()`, `q = create_ghz_state(d, n)`,
- `q = create_wstate(n)`, `qden = create_bes_horodecki_24_state()`,
- `qden = create_bes_horodecki_33_state()`,
- `qden = create_isotropic_state(p, d, n)`,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

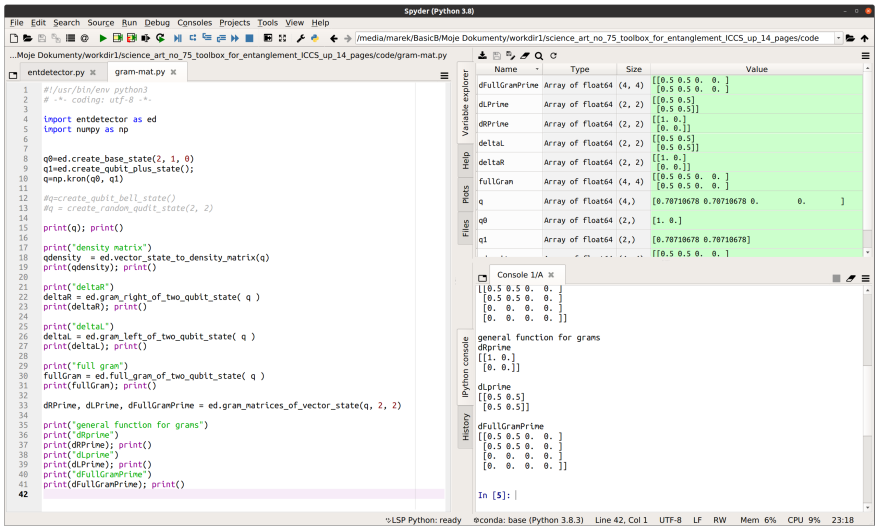

Fig. 1. Execution of an example calculating left, right, and full Gram matrices for the product system generated by the states $|0\rangle$ and $|+\rangle$. The example was run in Spyder 4.x editor, system distribution: Linux Ubuntu 20.04.01 LTS, environment Anaconda 4.x, Python version 3.8 64bit

where d stands for the dimension of a quantum unit, n represents the number of qudits, and p is a real value between 0 and 1. The variable q contains a vector state, and qden encloses a density matrix. The functions names are quite complex, but they describe the purpose of usage clearly.

The package offers basic functions for estimating entanglement's level, e.g. computing entropy, values of Concurrence and Negativity measures, calculating monotones for the systems of two (there also exist functions for three, and four) qubits:

- `v = entropy(qden)`, `v = concurrency(qden)`, `v = negativity(qden)`,
- `v = monotone_for_two_qubit_system(qden)`,

where qden represents a density matrix, and the result is assigned to the variable v. Types of arguments for presented functions are objects defined in the NumPy library, so it is easy to perform calculation joining functions from different packages (written in Python) dedicated to quantum computing, e.g. QuTIP.

The EntDetector includes a new tools like Gram matrices generation and a joined spectral-Schmidt decomposition:

- `tblsas = create_spectral_and_schmidt_data(qden, schmidt_shape)`,
- `right, left, full = gram_matrices_of_vector_state(v, d1, d2)`.

Naturally, we are not able to signal all implemented functions here. A current version of the package, directly related to this paper, is available at [22]. However,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

a future versions of the package and documentation are going to be published at main repository of our project in [21].

Remark 4. It should be emphasised once more that the main goal of the EntDetector package is to provide a user-friendly set of functions which allow analysing entanglement level with the help of known methods, e.g. the Negativity measure or Schmidt decomposition. It is also important to introduce some new methods, like density matrices expressed as Gramians what gives us a prospect of entanglement examination with the use of Gram volume and SaSD method, proposed in this work. The EntDetector should supplement other existing packages, like qiskit and QuTIP, with additional functions dedicated to entanglement analysis (e.g. SaSD tables, Gramian matrices).

3.2 Simple examples

The first example that we would like to present, is the Schmidt decomposition of an entangled state – more precisely: one of so-called EPR pairs: $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. To do it, we need to generate the mentioned state, perform the decomposition, and check if two Schmidt coefficients could be obtained. The code realising this task has the following form:

```
from entdetector import *

q = create_qubit_bell_state()
schmidt_shp=(2, 2)
s,e,f = schmidt_decomposition_for_vector_pure_state(q, schmidt_shp)
```

The function executing Schmidt decomposition requires introducing dimensions of the decomposition. The size of EPR vector is 4, so the decomposition's dimensions are 2×2 (given as a variable: schmidt_shp=(2, 2)). A reconstruction of quantum states is realised by a function:

```
qrebuild = reconstruct_state_after_schmidt_decomposition(s, e, f)
```

The Schmidt coefficients values are assigned to the variable `s`. The basis vectors are stored in `e` and `f`. Naturally, in the example with the EPR pair, we obtain two Schmidt coefficients equal to numeric value 0.70710678.

3.3 Spectral and Schmidt decomposition

We described the properties of spectral decomposition and the Schmidt decomposition in section 2.1. Let us generate the pure state q = q0 $\otimes$ qplus:

```
q0 = create_qubit_zero_state()
qplus = create_qubit_plus_state()
q = np.kron(q0, qplus)
```

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

where q0 = $|0\rangle$ and qplus = $|+\rangle$. Now, we can calculate both decompositions directly:

```
schmidt_shape=(2, 2)
qden = vector_state_to_density_matrix( q )
sas_tbl = create_sas_table_data(qden, schmidt_shape)
```

As the result, we obtain the following SaSD given in Eq. (9):

```
>>> print_sas_table( sas_tbl )
1.0  0.0 | 0.9999
1.0  0.0 | 0.0
1.0  0.0 | 0.0
1.0  0.0 | 0.0
```

A SaSD table for the Bell state $q = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is generated by:

```
q = create_qubit_bell_state()
qden = vector_state_to_density_matrix( q )
sas_tbl = create_sas_table_data(qden, schmidt_shape)
```

has the form:

```
>>> print_sas_table( sas_tbl )
0.70710678 0.70710678 | 0.9999
1.0        0.0        | 0.0
1.0        0.0        | 0.0
0.70710678 0.70710678 | 0.0
```

3.4 Decomposition of two-qubit isotropic state

SaSD tables allow analysing the isotropic state given as:

$$\rho = (p \cdot |\psi_+\rangle\langle\psi_+|) + (1-p)\frac{1}{d^2} \cdot (\mathbb{I}_d \otimes \mathbb{I}_d), \tag{27}$$

where in our case we assume that p is real and $-\frac{1}{d^2-1} \leq p \leq 1$. The parameter d is the dimension of a quantum unit utilized to construct the maximally entangled state $|\psi_+\rangle$ (and its density matrix $|\psi_+\rangle\langle\psi_+|$). The state ρ in general is entangled when $\frac{1}{d+1} < p \leq 1$, and in a contrary, separable if $-\frac{1}{d^2-1} \leq p \leq \frac{1}{d+1}$.

The SaSD table for the isotropic state with $d = 2$ and $p = 0$ is calculated by:

```
p=0.0
q = create_qubit_bell_state()
qdentmp = np.outer(q, q)
qden = (p * qdentmp) + ((1-p) * 0.25 * np.eye(4))
schmidt_shape=(2, 2)
sas_tbl = create_sas_table_data(qden, schmidt_shape)
```

and the result as SaSD table is:

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

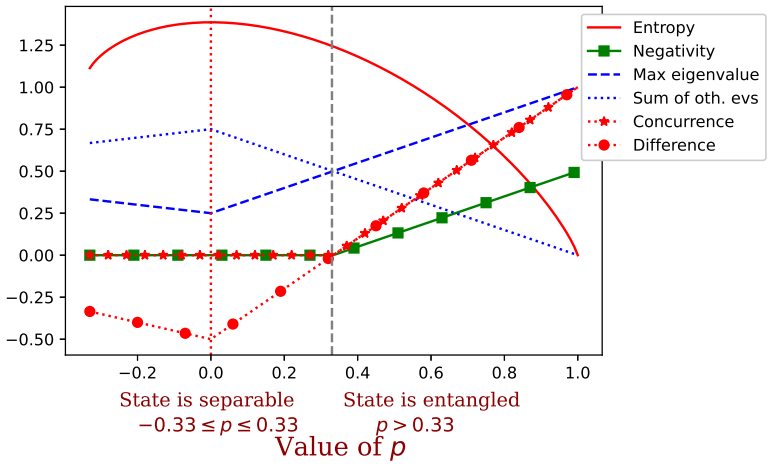

Fig. 2. The value of the maximal eigenvalue, the sum of others eigenvalues (denoted as sum of oth. evs), and also their difference for two-qubit isotropic state. The figure also shows the values of other quantum entanglement measures like Negativity and Concurrence. It is possible to point out that the Concurrence measure and the difference between the max eigenvalue and the sum of others eigenvalues are equal for $p \geq 0.33$

```
>>> print_sas_table( sas_tbl )
1.0 0.0 | 0.25
1.0 0.0 | 0.25
1.0 0.0 | 0.25
1.0 0.0 | 0.25
```

It should be noted that for parameter $p = 0$ the examined isotropic state takes a form of the maximally mixed state.

A SaSD table analysis enables entanglement detection in a given isotropic state for $d = 2$. Let us generate the exemplary tables for $p = 0.25$ and $p = 0.75$:

```
p=0.25                          p=0.75
0.7071 0.7071 | 0.5125          0.7071 0.7071 | 0.8125
0.9265 0.3760 | 0.1625          1.0    0.0    | 0.0625
0.9070 0.4209 | 0.1625          1.0    0.0    | 0.0625
0.8243 0.5661 | 0.1625          0.7071 0.7071 | 0.0625
```

The numeric analysis shows that we can point out the greatest eigenvalue correlated with the row having two non-zero coefficients what allows delineating the border between entangled and separable states. This phenomenon is depict on the plot in Fig. 2.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

4 Conclusions

In the article, we presented a package supporting work with qubit and qudit systems in terms of detecting entanglement in bipartite systems. The most important assumption was to provide convenient tools in the form of computational functions that perform the necessary decompositions, as well as entanglement detection functions.

In addition to providing support for known tools in the field of entanglement research, such as the Negativity and Concurrence measures, our package also offers new tools useful in the field of entanglement, such as the left and right Gramian of a given quantum state. EntDetector also offers an additional tool called SaSD. The information obtained with SaSD allows us, for example, to calculate the entropy for the studied quantum state.

The package has also the ability to examine the presence of entanglement in the indicated parts of a quantum register. The current procedure enables, for example, detecting entanglement, e.g. in graph states, and to verify whether a given examined state contains correctly entangled qubits.

Acknowledgments We would like to thank for useful discussions with the *Q-INFO* group at the Institute of Control and Computation Engineering (ISSI) of the University of Zielona Góra, Poland. We would like also to thank to anonymous referees for useful comments on the preliminary version of the paper. The numerical results were done using the hardware and software available at the "GPU μ-Lab" located at the Institute of Control and Computation Engineering of the University of Zielona Góra, Poland.

References

1. Abraham, H., et al.: QISKIT: An open-source framework for quantum computing (2019). https://doi.org/10.5281/zenodo.2562110
2. Bengtsson, I., Życzkowski, K.: Geometry of Quantum States: An Introduction to Quantum Entanglement. 2nd edn. Cambridge University Press, Cambridge, England (2017). https://doi.org/10.1017/9781139207010
3. Bennett, C.H., Bessette, F., Brassard, G., Salvail, L., Smolin, J.: Experimental quantum cryptography. Journal of Cryptology **5**, 3–28 (1992). https://doi.org/10.1007/BF00191318
4. Brassard, G., Braunstein, S.L., Cleve, R.: Teleportation as a quantum computation. Physica D: Nonlinear Phenomena **120**(1), 43 – 47 (1998). https://doi.org/10.1016/S0167-2789(98)00043-8
5. Einstein, A., Podolsky, B., Rosen, N.: Can quantum-mechanical description of physical reality be considered complete? Phys. Rev. **47**, 777–780 (1935). https://doi.org/10.1103/PhysRev.47.777
6. Gharibian, S.: Strong np-hardness of the quantum separability problem. Quantum Information and Computation **10**(3 & 4), 343–360 (2010)
7. Gühne, O., Tóth, G.: Entanglement detection. Physics Reports **474**(1), 1 – 75 (2009). https://doi.org/10.1016/j.physrep.2009.02.004

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9

8. Gielerak, R., Sawerwain, M.: A gramian approach to entanglement in bipartite finite dimensional systems: the case of pure states. Quantum Information and Computation **20**(13 & 14), 1081 – 1108 (2020). https://doi.org/10.26421/QIC20.13-1
9. Gielerak, R., Sawerwain, M.: in preparations (2021)
10. Gisin, N., Thew, R.: Quantum communication. Nature Photonics **1**, 165 – 171 (2007). https://doi.org/10.1038/nphoton.2007.22
11. Golub, G.H., van Loan, C.F.: Matrix Computations. 4th edn. Johns Hopkins University Press, Baltimore, Maryland, USA, (2013)
12. Gurvits, L.: Classical deterministic complexity of edmonds' problem and quantum entanglement. In: Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, STOC '03, pp. 10 – 19. Association for Computing Machinery, New York, NY, USA (2003). https://doi.org/10.1145/780542.780545
13. Harris, C.R., Millman, K.J., et al.: Array programming with NumPy. Nature **585**(7825), 357–362 (Sep 2020). https://doi.org/10.1038/s41586-020-2649-2
14. Horodecki, M., Horodecki, P., Horodecki, R.: Separability of mixed states: necessary and sufficient conditions. Physics Letters A **223**(1), 1 – 8 (1996). https://doi.org/10.1016/S0375-9601(96)00706-2
15. Horodecki, R., Horodecki, P., Horodecki, M., Horodecki, K.: Quantum entanglement. Rev. Mod. Phys. **81**, 865–942 (Jun 2009). https://doi.org/10.1103/RevModPhys.81.865
16. Johansson, J., Nation, P., Nori, F.: QuTiP: An open-source python framework for the dynamics of open quantum systems. Computer Physics Communications **183**(8), 1760 – 1772 (2012). https://doi.org/10.1016/j.cpc.2012.02.021
17. Johnston, N.: QETLAB: A MATLAB toolbox for quantum entanglement, version 0.9 (2016). https://doi.org/10.5281/zenodo.44637
18. Kolaczek, D., Spisak, B.J., Woloszyn, M.: The phase-space approach to time evolution of quantum states in confined systems: The spectral split-operator method. International Journal of Applied Mathematics and Computer Science **29**(3), 439–451 (2019). https://doi.org/10.2478/amcs-2019-0032
19. MATLAB: 9.7.0.1190202 (R2019b). The MathWorks Inc., Natick, Massachusetts (2018)
20. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. 10th edn. Cambridge University Press, Cambridge, England (2011)
21. Sawerwain, M., Wiśniewska, J., Wróblewski, M., Gielerak, R.: Github repository for EntDectector package (2021). https://github.com/qMSUZ/EntDetector
22. Sawerwain, M., Wiśniewska, J., Wróblewski, M., Gielerak, R.: Source code of EntDetector: entanglement detecting toolbox for bipartite quantum states (2021). https://doi.org/10.5281/zenodo.4643878
23. Tóth, G.: QUBIT4MATLAB V3.0: A program package for quantum information science and quantum optics for MATLAB. Computer Physics Communications **179**(6), 430–437 (2008). https://doi.org/10.1016/j.cpc.2008.03.007
24. Virtanen, P., Gommers, R., Oliphant, T., et al.: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods **17**, 261 – 272 (2020). https://doi.org/10.1038/s41592-019-0686-2

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_9