

Simulation of n -qubit quantum systems. III. Quantum operations [☆]

T. Radtke ^{*}, S. Fritzsche

Institut für Physik, Universität Kassel, Heinrich-Plett-Str. 40, D-34132 Kassel, Germany

Received 30 November 2006; received in revised form 23 February 2007; accepted 28 February 2007

Available online 20 March 2007

Abstract

During the last decade, several quantum information protocols, such as quantum key distribution, teleportation or quantum computation, have attracted a lot of interest. Despite the recent success and research efforts in quantum information processing, however, we are just at the beginning of understanding the role of entanglement and the behavior of quantum systems in *noisy* environments, i.e. for nonideal implementations.

Therefore, in order to facilitate the investigation of entanglement and decoherence in n -qubit quantum registers, here we present a revised version of the FEYNMAN program for working with quantum operations and their associated (Jamiołkowski) dual states. Based on the implementation of several popular decoherence models, we provide tools especially for the quantitative analysis of quantum operations. Apart from the implementation of different noise models, the current program extension may help investigate the fragility of many quantum states, one of the main obstacles in realizing quantum information protocols today.

Program summary

Title of program: Feynman

Catalogue identifier: ADWE_v3_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/ADWE_v3_0

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Licensing provisions: None

Operating systems: Any system that supports MAPLE; tested under Microsoft Windows XP, SuSe Linux 10

Program language used: MAPLE 10

Typical time and memory requirements: Most commands that act upon quantum registers with five or less qubits take ≤ 10 seconds of processor time (on a Pentium 4 processor with ≥ 2 GHz or equivalent) and 5–20 MB of memory. Especially when working with symbolic expressions, however, the memory and time requirements critically depend on the number of qubits in the quantum registers, owing to the exponential dimension growth of the associated Hilbert space. For example, complex (symbolic) noise models (with several Kraus operators) for multi-qubit systems often result in very large symbolic expressions that dramatically slow down the evaluation of measures or other quantities. In these cases, MAPLE's assume facility sometimes helps to reduce the complexity of symbolic expressions, but often only numerical evaluation is possible. Since the complexity of the FEYNMAN commands is very different, no general scaling law for the CPU time and memory usage can be given.

No. of bytes in distributed program including test data, etc.: 799 265

No. of lines in distributed program including test data, etc.: 18 589

Distribution format: tar.gz

Reasons for new version: While the previous program versions were designed mainly to create and manipulate the state of quantum registers, the present extension aims to support quantum operations as the essential ingredient for studying the effects of noisy environments.

Does this version supersede the previous version: Yes

Nature of the physical problem: Today, entanglement is identified as *the* essential resource in virtually all aspects of quantum information theory. In most practical implementations of quantum information protocols, however, decoherence typically limits the lifetime of entanglement. It is therefore necessary and highly desirable to understand the evolution of entanglement in noisy environments.

[☆] This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

^{*} Corresponding author.

E-mail addresses: t.radtke@physik.uni-kassel.de (T. Radtke), s.fritzsche@physik.uni-kassel.de (S. Fritzsche).

Method of solution: Using the computer algebra system MAPLE, we have developed a set of procedures that support the definition and manipulation of n -qubit quantum registers as well as (unitary) logic gates and (nonunitary) quantum operations that act on the quantum registers. The provided hierarchy of commands can be used interactively in order to simulate and analyze the evolution of n -qubit quantum systems in ideal and nonideal quantum circuits.

© 2007 Elsevier B.V. All rights reserved.

PACS: 03.67.Lx; 03.65.Ud; 03.75.Gg

Keywords: Quantum register; Quantum operation; Channel; Decoherence

1. Introduction

During the last decade, the field of quantum computation and information has become a rapidly growing research area with applications ranging from quantum teleportation [1] and quantum cryptography [2] to superdense coding [3] and efficient quantum computations [4,5]. These applications are famous examples of quantum information protocols, i.e. procedures in which one acts on quantum states in order to achieve tasks that are not possible in classical information processing. In all these protocols, entanglement has been identified as the central resource that is necessary in order to ‘exploit’ the remarkable features of quantum information processing [6,7].

Although several of the concepts, which were suggested only theoretically at the beginning of these developments, have meanwhile been implemented with great success in the laboratory [8–10], the fragility of quantum entanglement is still a limiting factor for many practical realizations. Indeed, the decoherence in quantum states, i.e. their unwanted interaction with the environment, is a major challenge in all applications. To overcome the difficulties due to the decoherence in the phases and/or amplitudes of complex quantum states, a number of active and passive methods have been suggested to protect and recover the contained quantum information [11–13].

In dealing with the interaction of (open) quantum systems with their environment, especially the concept of quantum operations (or completely positive maps) has become a valuable tool to describe the errors and decoherence effects. In contrast to the frequently applied Markovian approach of master equations [14,15], the use of quantum operations is rather general and does not necessarily imply assumptions about the strength of the system-environment coupling. Moreover, quantum operations are suitable to describe discrete error models which are the starting point for quantum error correction methods.

To facilitate simulations of the evolution and behavior of n -qubit quantum systems, we have recently introduced the FEYNMAN program, a simple quantum simulator within the framework of the general-purpose computer algebra system MAPLE. With this program, our intention is to provide an easily extendible set of symbolic and numerical utilities, which help shorten some of the typical calculations as they appear frequently in the simulation of quantum registers. In a first version [16], therefore, we provided the tools for creating and manipulating quantum registers, for applying quantum gates on quantum registers, or for analyzing their properties with regard to entanglement, entropy, or their distance from other quantum states [17]. In the present work, we extend these tools to enable the user to deal also with quantum operations. Specifically, the new commands provide a simple access to some frequently used decoherence models and enable also a quantitative analysis by implementing several measures for quantum operations. Moreover, by using the so-called ‘Jamiołkowski dual state’ representation of a quantum operation, many tools for the analysis of quantum states can now be used also for quantum operations. We therefore hope that the present extension will simplify the analysis of decoherence phenomena in n -qubit quantum systems and will make the FEYNMAN program an attractive toolbox for applications in the field of quantum computation and information.

In the next section, we briefly provide some theoretical background on the concept and use of quantum operations, together with the nomenclature as applied in the FEYNMAN program below. Apart from the formal properties of these operations, this includes a short discussion on their relation to the (dual) states due to the Jamiołkowski isomorphism as well as on several measures for characterizing the action of these operations. Based on this background, Section 3 later describes how quantum operations are implemented in the FEYNMAN program and which new commands are available to work with them. Section 4 then provides four short examples that illustrate the interactive work with the program and where some of the new features are shown in more detail. In Section 5, finally, a brief summary and outlook on the possible further development of the program is given.

2. Theoretical background

Positive and completely positive maps play a very important role in the context of quantum information theory, concerning especially the investigation of decoherence and entanglement in quantum systems. For instance, based on a detailed analysis of positive (but not completely positive) maps, several criteria have been developed in the past in order to be able to decide whether a quantum state is entangled or separable [18]. Several of such separability criteria for pure and mixed quantum states were implemented already in our previous version of the FEYNMAN program [17]. In this section, we now focus on the description of decoherence phenomena using completely positive maps, briefly referred to as quantum operations. Here, emphasis is placed on the basic defin-

itions and notations as implemented in the current program extension. For a more detailed introduction to the dynamics and noisy evolution of *open* quantum systems we refer the reader to Refs. [14,15].

2.1. Evolution of closed quantum systems

For *closed* quantum systems, the time evolution of a pure state (wavefunction) $|\psi\rangle$ is often governed by a Schrödinger equation

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H(t) |\psi(t)\rangle \quad (1)$$

with the (possibly time-dependent) Hamiltonian $H(t)$. More generally, closed systems are known to follow a *unitary* time-evolution operator $U(t_0, t)$ that connects some initial state $|\psi(t_0)\rangle$ at time t_0 to the states $|\psi(t)\rangle$ at any other time t by

$$|\psi(t)\rangle = U(t, t_0) |\psi(t_0)\rangle \quad (2)$$

with

$$U(t, t_0) = \hat{T}_{\leftarrow} \exp \left[-\frac{i}{\hbar} \int_{t_0}^t dt' H(t') \right] \quad (3)$$

and $U^\dagger U = U U^\dagger = I$. In the time evolution operator (3), \hat{T}_{\leftarrow} denotes the chronological time-ordering, i.e. an operator that orders all products of time-dependent operations in a such way that the operators for the earliest times come to the right and those for later times follow to the left. For the special case of an *isolated* (physical) system where the Hamiltonian is time independent, Eq. (3) simplifies to

$$U(t, t_0) = \exp \left[-\frac{i}{\hbar} H(t - t_0) \right]. \quad (4)$$

Similarly, the evolution of a mixed quantum state $\rho(t) = \sum_i p_i |\psi_i(t)\rangle \langle \psi_i(t)|$ is given by

$$\rho(t) = U(t, t_0) \rho(t_0) U^\dagger(t, t_0). \quad (5)$$

2.2. Subsystem evolution and quantum operations

In many cases, a physical system S cannot be considered as *isolated* or *closed* owing to its interaction with some environment E . In these cases, the evolution of the system (without its environment) cannot be described by a Hamiltonian operator any more. Still, the general formulas from the previous section remain valid for the *composite* system SE if this is taken to be closed again. Assuming that the principal system S and the environment E are initially uncorrelated and that the environment was in some pure (basis) state $|e_0\rangle$ at that time (which is always possible by performing a purification), the general unitary evolution of the whole system SE can be written as

$$\rho_{SE}(t) = U_{SE}(t, t_0) (\rho_S(t_0) \otimes |e_0\rangle \langle e_0|) U_{SE}^\dagger(t, t_0). \quad (6)$$

The evolution of the subsystem S is then obtained by tracing over the degrees of freedom of the environment

$$\begin{aligned} \rho'_S &= \mathcal{E}(\rho_S) = \text{Tr}_E [U_{SE}(\rho_S \otimes |e_0\rangle \langle e_0|) U_{SE}^\dagger] \\ &= \sum_k \langle e_k | U_{SE}(\rho_S \otimes |e_0\rangle \langle e_0|) U_{SE}^\dagger | e_k \rangle \\ &= \sum_k E_k \rho_S E_k^\dagger, \end{aligned} \quad (7)$$

where $\{|e_k\rangle\}$ denotes an orthonormal basis for the environment and where, for the sake of simplicity, we have omitted the explicit time-dependence of the evolution operators and density matrices. Following the notation of the textbook by Nielsen and Chuang [6], we shall call below the mapping

$$\rho \mapsto \mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger, \quad \sum_k E_k^\dagger E_k \leq I \quad (8)$$

a *quantum operation* with the *operation elements* $E_k = \langle e_k | U_{SE} | e_0 \rangle$, which are acting only on the subsystem S . In the literature, such quantum operations are also known as quantum channels, superoperators [7] or completely positive maps (see below), while the operation elements E_k are sometimes referred to as Kraus operators [19]. Note, however, that the description of a quantum operation in terms of its operation elements is not *unique* as seen from the freedom in choosing an orthonormal basis $\{|e_k\rangle\}$ for

the environment. In particular, two representations $\{E_1, \dots, E_m\}$ and $\{F_1, \dots, F_n\}$ of a quantum operation are considered to be the same if they are related by an $m \times m$ unitary matrix (u_{ij})

$$E_i = \sum_j u_{ij} F_j. \quad (9)$$

Here, one has to append zero operators, if necessary, in order to achieve $m = n$. Despite this unitary freedom in the operator-sum representation, there exists a canonical decomposition with a minimum number of operation elements, cf. Section 2.3.1.

In order to describe a physically meaningful evolution of a (sub)system ρ , we require a quantum operation \mathcal{E} of the general form (8) to be a linear operator that maps valid density operators onto valid density operators again. This requirement implies that a quantum operation \mathcal{E} has to possess the following properties:

- (i) \mathcal{E} is hermiticity-preserving: $\rho = \rho^\dagger \Rightarrow \mathcal{E}(\rho) = \mathcal{E}(\rho)^\dagger$,
- (ii) \mathcal{E} is trace-preserving: $\text{Tr}(\rho) = 1 \Rightarrow \text{Tr}(\mathcal{E}(\rho)) = 1$ (which requires $\sum_k E_k^\dagger E_k = I$),
- (iii) \mathcal{E} is positivity-preserving: $\rho \geq 0 \Rightarrow \mathcal{E}(\rho) \geq 0$,
- (iv) \mathcal{E} is *completely* positive: for any extension \mathcal{H}_E of the initial state space \mathcal{H}_S , we require that $\mathcal{H}_S \otimes \mathcal{H}_E \ni \rho_{SE} \geq 0 \Rightarrow [\mathcal{E}_S \otimes \mathcal{I}_E](\rho_{SE}) \geq 0$, independent of the particular dimension of \mathcal{H}_E .

While the first three conditions (i)–(iii) guarantee that the general properties of a density matrix apply also for the final density matrix $\mathcal{E}(\rho)$, condition (iv) might appear less obvious at the first glance. Still, this condition is physically reasonable since we require not only the *local* evolution $\rho'_S = \mathcal{E}_S(\rho_S)$ of some subsystem S to be positivity-preserving but also any *combined* evolution $\rho'_{SE} = [\mathcal{E}_S \otimes \mathcal{I}_E](\rho_{SE})$ which acts trivially on \mathcal{H}_E . Otherwise, such a *combined* evolution could lead to a joint state ρ'_{SE} that is not positive and, hence, not a valid density matrix. A well-known counterexample of a positive but not *completely* positive map is the transposition operator T , which is also an example for using of a positive map as separability criterion [18]. Although we have $\rho \geq 0 \Rightarrow T(\rho) = \rho^T \geq 0$, the same is in general not true if the operator T is applied only to the subsystem A (or B) of some composite system AB . In fact, the resulting matrix $\rho^{TA} = [T_A \otimes \mathcal{I}_B](\rho_{AB})$ is then not always a valid density matrix, since it may have negative eigenvalues. This can easily be verified, for example, for the maximally entangled two-qubit state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ [6].

A popular example of a valid quantum operation and, hence, for a completely positive trace-preserving map is the so-called (single-qubit) *phase damping* or dephasing channel, which is one of the most important decoherence models. It describes a diffusive scattering interaction of the qubit with its environment [6,7]. A possible representation of this phase damping channel in terms of operation element is given by

$$\mathcal{E}_{\text{PD}}(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger \quad (10)$$

with

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix}, \quad E_1 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{p} \end{bmatrix}, \quad (11)$$

and where $p = 0, \dots, 1$ can be interpreted as a scattering probability. For $p \rightarrow 1$, the quantum operation causes the off-diagonal elements of the density matrix ρ (of a single-qubit system) to vanish. Therefore, this channel will asymptotically destroy all phase information which is crucial to quantum information processing. Note, however, that there exist also other representations of the phase damping channel owing to the unitary freedom (9) in the operator-sum representation (8). One example of such a unitarily related representation is given by the *phase-flip* channel with the operation elements

$$E'_0 = \sqrt{1-p'} I = \sqrt{1-p'} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad E'_1 = \sqrt{p'} \sigma_z = \sqrt{p'} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (12)$$

One can show that, for $p' = (1 - \sqrt{1-p})/2$, the two representations are unitarily equivalent. Among other pre-defined quantum operations, the phase-damping channel and the phase-flip channel can be accessed via the `Feynman_quantum_operation()` command (see also Table 3).

Finally, in order to avoid confusion about the term *decoherence*, let us note that it referred originally only to the purely quantum mechanical dephasing model from above. Today, however, it is used also for other types of interaction with the environment in the literature.

2.3. Duality between quantum operations and states

Having defined a quantum operation and its fundamental properties in the previous section we recall now an interesting and powerful connection between quantum operations and quantum states. In particular, for every quantum operation \mathcal{E} that acts on

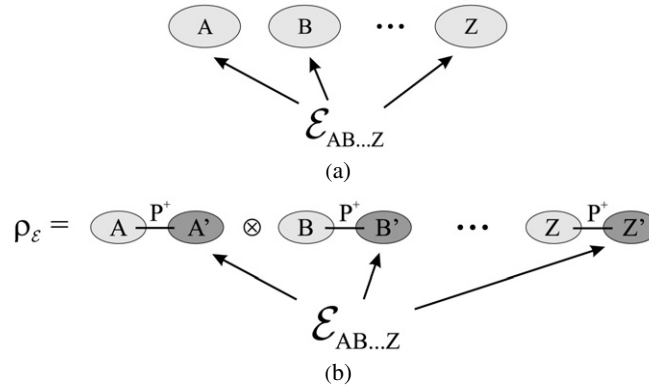


Fig. 1. (a) Given is a quantum operation $\mathcal{E}_{AB\dots Z}$ that acts on the subsystems A, B, \dots, Z (which may each consist of one or several qubits). (b) The corresponding dual state $\rho_{\mathcal{E}}$ is obtained by letting $\mathcal{E}_{AB\dots Z}$ act only on the copied subsystems A', B', \dots, Z' , where each subsystem A' (B', \dots) is in the maximally entangled pure state $P^+ = |\Phi^+\rangle\langle\Phi^+|$ with its copy A' (B', \dots).

Table 1

Some important relations between the properties of quantum operations and the properties of their corresponding dual states [21,22]

Quantum operation \mathcal{E}	Quantum state $\rho_{\mathcal{E}}$
Completely positive	positive
Trace-preserving	positive and $\text{Tr}'_{\mathcal{H}}(\rho_{\mathcal{E}}) = \frac{1}{d_{\mathcal{H}}}\mathbb{1}$
Hermiticity-preserving	hermitian
Factorizable (or pure), i.e. \mathcal{E} can be written as $\mathcal{E}(\rho) = E_1\rho E_1^\dagger$	pure
Unital (or bistochastic)	$\text{Tr}_{\mathcal{H}}(\rho_{\mathcal{E}}) = \frac{1}{d_{\mathcal{H}}}\mathbb{1}$ and $\text{Tr}'_{\mathcal{H}}(\rho_{\mathcal{E}}) = \frac{1}{d_{\mathcal{H}}}\mathbb{1}$
Entanglement breaking	separable

a d -dimensional state space \mathcal{H} , there exists an associated $2d$ -dimensional state $\rho_{\mathcal{E}}$ in the set of the density matrices $\mathcal{D}(\mathcal{H} \otimes \mathcal{H}')$. The so-called dual state $\rho_{\mathcal{E}}$ reflects the properties of the original quantum operation. This duality is known as the Jamiołkowski isomorphism [20–22] between hermiticity-preserving linear maps and hermitian matrices.

In the single-party setting, i.e. when we do not distinguish between different subsystems in the d -dimensional state space on which \mathcal{E} acts, we define the dual state $\rho_{\mathcal{E}} \in \mathcal{D}(\mathcal{H} \otimes \mathcal{H}')$ so that the quantum operation \mathcal{E} acts locally on one subsystem of the maximally entangled pure state $|\Phi^+\rangle$ in the extended space $\mathcal{H} \otimes \mathcal{H}'$ as follows

$$\rho_{\mathcal{E}} = (\mathcal{I}_{\mathcal{H}} \otimes \mathcal{E}_{\mathcal{H}'})|\Phi^+\rangle\langle\Phi^+| \quad \text{with } |\Phi^+\rangle = \frac{1}{\sqrt{d_{\mathcal{H}}}} \sum_{i=1}^{d_{\mathcal{H}}} |i\rangle_{\mathcal{H}}|i\rangle'_{\mathcal{H}}. \quad (13)$$

In the multipartite case, we assume that a given quantum operation \mathcal{E} acts on an n -partite state space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B \otimes \dots \otimes \mathcal{H}_Z$ with the dimensions d_k . Similarly to Eq. (13), the dual state is now defined as

$$\rho_{\mathcal{E}} = (\mathcal{I}_{\mathcal{H}} \otimes \mathcal{E}_{\mathcal{H}'})P^+ \quad (14)$$

so that \mathcal{E} acts only on the copy in $\mathcal{H}' = \mathcal{H}_{A'} \otimes \mathcal{H}_{B'} \otimes \dots \otimes \mathcal{H}_{Z'}$. The state P^+ is now given as the tensor product of the maximally entangled states in each composite subsystem AA', BB', \dots, ZZ' : $P^+ = P^+_{AA'} \otimes P^+_{BB'} \otimes \dots \otimes P^+_{ZZ'}$ where $P^+_{kk'} = |\Phi^+_{kk'}\rangle\langle\Phi^+_{kk'}|$ and $|\Phi^+_{kk'}\rangle = \frac{1}{\sqrt{d_k}} \sum_{i=1}^{d_k} |ii\rangle_{kk'}$ (cf. [23]). An illustration of the multipartite version of the duality between quantum operations and states is shown in Fig. 1.

The duality between quantum operations and quantum states is particularly useful as it allows one to characterize a quantum operation in terms of the well-known properties of its corresponding state. Some of those relations are given in Table 1. In the FEYNMAN package, the dual state of a quantum operation can be calculated with the help of the Feynman_qoperation_representation() command. Moreover, the concept of the dual state is also used by the Feynman_qoperation_type() command to check several properties of quantum operations and by the Feynman_measures() command in order to evaluate a variety of measures for quantum operations (see also Table 4).

2.3.1. Related representations of quantum operations

In addition to the operator-sum representation (or Kraus representation) and the description via the corresponding dual state, there are also other (related) representations of a quantum operation which are used in the literature and in the FEYNMAN program.

The first alternative representation is the ‘superoperator’ matrix L which emphasizes the fact that a quantum operation \mathcal{E} is a linear map (i.e. an operator) acting on density matrices (operators). Therefore, a quantum operation $\rho' = \mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger$ can be

equivalently characterized in the ‘vectorized’ notation of the linear superoperator

$$\text{vec}(\rho') = L \text{vec}(\rho) \quad \text{where } L = \sum_i E_i^* \otimes E_i, \quad (15)$$

where $\text{vec}(A)$ stands for the (column) vector that is obtained by appending all columns of the $m \times n$ matrix A to its first column so that we have a single column of dimension mn (see Ch. 4 in [24]). As follows from the definition (15), the resulting superoperator matrix L has the dimension $d^2 \times d^2$ if the operation elements are of dimension $d \times d$.

Closely related to the superoperator matrix L is the Choi matrix (or ‘dynamical matrix’) D being defined as the ‘realigned’ superoperator matrix L : $D = \mathcal{R}(L)$. The ‘realignment’ operation $\mathcal{R}(L)$ is a well-defined rearrangement of the matrix entries of L (see [25] and the FEYNMAN program manual). Due to the connection between the spectral decomposition of the Choi matrix and the operator-sum representation of a quantum operation, the Choi matrix also provides a canonical operator-sum decomposition where the minimum number of operation elements is determined by the rank of the Choi matrix. We have

$$D = \sum_{i=1}^r d_i |\chi_i\rangle\langle\chi_i| \Rightarrow \mathcal{E}(\rho) = \sum_{i=1}^r E_i \rho E_i^\dagger \quad \text{where } E_i = \sqrt{d_i} \tilde{\chi}_i, \quad (16)$$

where $\tilde{\chi}_i$ is the matrix that results from the inverse $\text{vec}(\cdot)$ operation and $r = \text{Rank}(D)$. These canonical operation elements are unique (up to reordering).

In addition to the connection to the operator-sum representation, the Choi matrix is also related to the Jamiołkowski dual state $\rho_{\mathcal{E}} = D/d$ where $d = \dim(\mathcal{H})$ is the dimension of the space on which the quantum operation \mathcal{E} acts, i.e. $d = 2^n$ for an n -qubit channel. Note, however, that there are slightly different definitions of L and D based on the different versions of the $\text{vec}(\cdot)$ operation [22,26].

2.4. Measures for quantum operations

In many situations it is useful to have, for example, a distance measure to compare some quantum operation describing an actual physical process to an idealized process (e.g., a quantum gate) [27]. By using the duality (14) between quantum operations and states, several distance measures for states can be translated to quantum operations in a straightforward way. Symbolically, we can write

$$\tilde{D}(\mathcal{E}, \mathcal{F}) = D(\rho_{\mathcal{E}}, \rho_{\mathcal{F}}), \quad (17)$$

where D denotes one of the well-known (distance) measures for states, such as the trace distance, fidelity, Bures distance and others which have different advantages and problems depending on the application [27].

In addition to the distance measures, there exist, of course, several quantities that frequently serve as a measure how of well a quantum operation preserves physical properties or resources like the purity of an input state or its entanglement [28,33]. Such kind of measures are often applied to characterize the effects of decoherence and noise acting on a quantum register. We do not discuss these measures here but refer to an overview of the implemented measures for quantum operations which is given in Table 4.

3. Implementation and further extensions to the FEYNMAN program

3.1. Overview

Our main motivation for developing the FEYNMAN program is to provide a toolbox for the simulation of n -qubit quantum systems (quantum registers) within the framework of MAPLE, i.e. for a general-purpose computer algebra system with widespread use in science and engineering. For this purpose, the FEYNMAN program has been organized as a *module* (package) for MAPLE, containing a hierarchy of currently about 60 procedures at quite different levels of complexity. Apart from various subprocedures, which remain *hidden* to the user, the *main* commands can be used for interactive work and as language elements in order to build-up new commands at some higher level of the hierarchy. Although a number of similar, algebraic, programs exist also for MATHEMATICA [29], in most of the other cases, emphasis in developing ‘qubit simulators’ has been placed on purely numerical studies, cf. Ref. [30]. In contrast, the FEYNMAN program is designed for performing either symbolic *and/or* numerical computations. Moreover, making use of the large number of MAPLE’s built-in functions, only a minimum of programming effort is usually required in order to extend the program and adapt it to user-specific needs.

In a first version of the FEYNMAN program [16], the basic data structures for the simulation of n -qubit quantum systems were established, referring especially to the three data types `qbit()`, `qregister()` and `qoperator()`, respectively. These structures (procedures) are utilized for keeping related information together and to facilitate the data handling *with* and *within* the program. Moreover, a number of commands were introduced in this version in order to act upon these data structures, e.g., by applying some pre-defined or user-defined quantum gates or for performing standard applications, such as the tensor product of two or more quantum registers

Table 2

New (or extended) commands of the FEYNMAN program as accessible by the users. A detailed description of all commands is provided in the manual `Feynman-commands.pdf` which is distributed together with the code

Command	Short explanation
<code>Feynman_compose()</code>	Computes the <i>composition</i> of two (or more) quantum operations, i.e. the successive application of two or more quantum operations. In contrast, the tensor product of two quantum operations is supported by the command <code>Feynman_product("Kronecker", ...)</code> .
<code>Feynman_measures()</code>	Evaluates several measures for quantum operations as summarized in Table 4.
<code>Feynman_qoperation_invariants()</code>	Returns the fixed points of a quantum operation, i.e. those matrix operators that are invariant under the considered quantum operation.
<code>Feynman_qoperation_representation()</code>	Provides several representations of a quantum operation as outlined in Section 2.3.1.
<code>Feynman_qoperation_rank()</code>	Returns the (Choi) rank of a quantum operation.
<code>Feynman_qoperation_type()</code>	Tests whether a quantum operation has some given property, such as collective (that is invariant under qubit permutation), unital and others, or if this is not the case.
<code>Feynman_quantum_operation()</code>	Provides access to a number of pre-defined and frequently applied single-qubit and multi-qubit quantum operations (channels).

or the computation of reduced density operators. In addition, some simple visualization tools were provided in order to ‘display’ the state of quantum registers by using probability plots or the Bloch vector representation of single qubits. In a second version, later, we focused mainly on the entanglement properties of quantum states by implementing several separability criteria and ‘measures’ for entanglement. Most of these measures can be accessed by means of the command `Feynman_measures()`, together with a number of other, frequently applied quantities concerning the distance between or the entropy of quantum states.

With the present extension of the program, we now concentrate on the definition and application of *quantum operations*, i.e. on completely positive and trace-preserving maps (CPT maps) that can be written in the form of Eq. (8). In fact, the concept of the quantum operations provides a very general tool in order to describe the (typically nonunitary) evolution of quantum register if they interact with their environment. The use of quantum operations hereby emphasizes the *discrete* changes of the state of a quantum system, in contrast to the *continuous-time* description by means of master equations [6,14]. Note that for every time-dependent quantum state $\rho(t)$, there exists always a description of its evolution in terms of an operator-sum representation which may, however, depend on the initial state of the system [31,32].

3.2. Implementation of quantum operations

In the FEYNMAN program, the data structure `qoperation()` is utilized to deal with quantum operations, as introduced already in the first version of the program [16]. However, in order to support the work with such operations and operator-sum representations, seven *main* commands have been added (or significantly extended) to the FEYNMAN program. A brief summary of these additional commands can be found in Table 2 while, for all further details, we refer the reader to the manual `Feynman-commands.pdf` which is distributed together with the code.

The data structure `qoperation()` is used to describe some quantum operation for a given ‘size’ of the system, i.e. for a given number of qubits; this data object either contains a *keyword* description of a quantum operation or simply a list with the explicit matrix representations of the corresponding operation elements. The use of keywords (together with a specified number of additional parameters) allows for a compact notation for quantum operations which can later be evaluated on demand by invoking the command `Feynman_quantum_operation()`. This procedure knows not only a number of pre-defined quantum operations, as they appear frequently in the literature, but supports also their ‘distribution’ in the operation elements, if applied to quantum systems larger (in the number of qubits) than the corresponding noise model. For example, the command `Feynman_quantum_operation()` knows the popular (single-qubit) dephasing and amplitude damping channels as well as a few simple n -qubit generalizations of these noises [6,7]. An overview about the currently *pre-defined* quantum operations is given in Table 3.

In fact, the implementation and use of the data structure `qoperation()` and the command `Feynman_quantum_operation()` is analogue to the implementation of quantum gates, for which we have `qoperator()` to provide the data structure and the command `Feynman_quantum_operator()` for the explicit evaluation of this structure. In both cases, therefore, the data structures can be considered as language elements that basically store the unevaluated argument options, while `Feynman_quantum_operation()` and `Feynman_quantum_operator()` are utilized to evaluate the language elements to explicit matrix representations. Apart from defining and evaluating quantum operations, the FEYNMAN program now supports also compositions and tensor products of quantum operations by means of the commands `Feynman_compose()` and `Feynman_product()`, respectively.

In addition to the commands above, which enable the user to create, combine and apply a `qoperation()`, we also provide tools for the—qualitative and quantitative—analysis of quantum operations. To this end, the command `Feynman_qoperation_type()` has been added and the procedure `Feynman_measures()` has been extended substantially. The command `Feynman_qoperation_type()` can be used to test a `qoperation()` for being of a given type (property), for instance, of being trace-preserving, unital ($\mathcal{E}(\mathbb{1}) = \mathbb{1}$),

Table 3
List of pre-defined quantum operations that are implemented by the command `Feynman_quantum_operation()`. A more detailed description of all argument options of this command is provided in the manual `Feynman-commands.pdf`

Argument option	to return the list of operation elements of
<i>Single-qubit quantum operations</i>	
("Pauli", p_0, p_1, p_2, p_3)	$\mathcal{E}(\rho) = p_0\rho + \sum_{i=1}^3 p_i\sigma_i\rho\sigma_i$, where $p_i \geq 0$, $\sum_i p_i = 1$ and σ_i are the Pauli matrices.
("depolarize", p)	$E_0 = \sqrt{1 - \frac{3}{4}p}I, E_1 = \sqrt{\frac{p}{4}}X, E_2 = \sqrt{\frac{p}{4}}Y, E_3 = \sqrt{\frac{p}{4}}Z$
("phase damping", p)	$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix}, E_1 = \sqrt{p}\sigma_- \sigma_+ = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{p} \end{bmatrix}$
("phase flip", p)	$E_0 = \sqrt{1-p}I, E_1 = \sqrt{p}Z$
("bit flip", p)	$E_0 = \sqrt{1-p}I, E_1 = \sqrt{p}X$
("bit+phase flip", p)	$E_0 = \sqrt{1-p}I, E_1 = \sqrt{p}Y$
("amplitude damping", p)	$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix}, E_1 = \sqrt{p}\sigma_+ = \begin{bmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{bmatrix}$
("generalized amplitude damping", p, γ) or ("GAD", p, γ)	$E_0 = \sqrt{1-\gamma} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix}, E_1 = \sqrt{1-\gamma} \begin{bmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{bmatrix},$ $E_2 = \sqrt{\gamma} \begin{bmatrix} \sqrt{1-p} & 0 \\ 0 & 1 \end{bmatrix}, E_3 = \sqrt{\gamma} \begin{bmatrix} 0 & 0 \\ \sqrt{p} & 0 \end{bmatrix}$
<i>n-qubit quantum operations</i>	
("individual amplitude damping", p, n)	Amplitude damping that acts only upon one out of n qubits at a time. For this quantum operation, the Kraus operators E_i have the tensor product structure
("individual phase flip", p, n)	$E_i = \sqrt{p/n}\sigma_+^{(i)} = \sqrt{p/n}I \otimes \dots \otimes \sigma_+ \otimes \dots \otimes I$ and $E_0 = \sqrt{I - \sum_{i=1}^n E_i^\dagger E_i}$. Individual phase flip that acts only upon one out of n qubits at a time. The Kraus operators E_i have a the tensor product structure
("fully correlated amplitude damping", p, n)	$E_i = \sqrt{p/n}\sigma_z^{(i)} = \sqrt{p/n}I \otimes \dots \otimes \sigma_z \otimes \dots \otimes I$ and $E_0 = \sqrt{I - \sum_{i=1}^n E_i^\dagger E_i}$. Correlated amplitude damping that acts upon all n qubits simultaneously. The two Kraus operators are given by $E_0 = \sqrt{I - E_1^\dagger E_1}$ and $E_1 = \sqrt{p}\sigma_+^{\otimes n}$.
("fully correlated phase flip", p, n)	Correlated phase flip operation that acts upon all n qubits simultaneously. The two Kraus operators are given by $E_0 = \sqrt{1-p}I^{\otimes n}$ and $E_1 = \sqrt{p}\sigma_z^{\otimes n}$.

or *collective*, that means invariant under the permutation of the qubit indices. In contrast, the command `Feynman_measures()` facilitates the computation of several measures, such as the entanglement fidelity or the distance of two quantum operations, which is defined via the duality between quantum operations and states, see Section 2.3 for details. In Section 4.2, moreover, we shall demonstrate how the command `Feynman_is_separable()` can be utilized to investigate the separability properties of a quantum operation, i.e. we demonstrate the connection between a quantum operation's ability to break the entanglement of arbitrary initial states and the entanglement of the quantum operation's dual state.

3.3. Further changes to the program

In addition to the program extensions, which are necessary in order to deal with quantum operations, some minor changes and improvements have been made in the FEYNMAN program. However, not much need to be said here about these modifications of the code. To facilitate numerical studies, we introduced the global variable `Feynman_precision` which defines a numerical noise threshold, e.g., when checking the equality of two objects or deciding if a separability criterion is fulfilled. For example, if the user wants to check whether a quantum operation (which is defined by symbolic or numeric operation elements E_i) is trace-preserving or not then the program calculates $M = \sum_i E_i^\dagger E_i$. Then, provided that M can be finally evaluated to a numeric matrix, the program checks if M equals the identity matrix. In the case of a symbolic matrix M that still contains variables, typically no decision can be made and the program proceeds without further message. In the numeric case, each matrix element M_{ij} may deviate from the expected value in the order of the `Feynman_precision` value before the program considers the quantum operation not to be trace-preserving. This tolerance is to avoid error messages that would occur otherwise due to rounding errors. A list of the user-accessible FEYNMAN commands that are affected by the `Feynman_precision` variable is given in the manual.

For the interactive work with the FEYNMAN procedures, we recommend the user to increase MAPLE's `Digits` variable to, say, `Digits := 20`, at the beginning of the session. By default, the `Feynman_precision` variable is set to 10^{-15} . Of course, both values can be changed manually by the user at any time during program execution.

3.4. Program distribution

As with the previous versions, the FEYNMAN program is distributed via the CPC program library as an compressed archive file that contains the FEYNMAN library file, the source code as well as the manual `Feynman-commands.pdf`. In addition, this

Table 4

List of measures that are implemented (additionally) by the command `Feynman_measures()` in order to deal with quantum operations. A more detailed description of all argument options of this command is provided in the manual `Feynman-commands.pdf`

Argument option	Explanation
("J distance", <code>qoperation\mathcal{E}</code> , <code>qoperation\mathcal{F}</code>)	Jamiołkowski distance $D_J(\mathcal{E}, \mathcal{F})$ which is equivalent to the trace distance $D_{\text{Tr}}(\rho_{\mathcal{E}}, \rho_{\mathcal{F}})$ between the corresponding dual states [27].
("J fidelity", <code>qoperation\mathcal{E}</code> , <code>qoperation\mathcal{F}</code>)	Jamiołkowski fidelity $F_J(\mathcal{E}, \mathcal{F}) = F(\rho_{\mathcal{E}}, \rho_{\mathcal{F}})$, i.e. the fidelity of the corresponding dual states [27].
("C-distance", <code>qoperation\mathcal{E}</code> , <code>qoperation\mathcal{F}</code>)	C-distance $C(\mathcal{E}, \mathcal{F}) = \sqrt{1 - F(\rho_{\mathcal{E}}, \rho_{\mathcal{F}})}$ [27].
("decoherence norm", <code>qoperation\mathcal{E}</code>)	Decoherence norm as given by the Hilbert–Schmidt norm of the associated ‘superoperator’ L : $N(\mathcal{E}) = \ L\ _{HS} = \sqrt{\text{Tr} L^\dagger L}$ [22].
("process purity", <code>qoperation\mathcal{E}</code>)	Process purity $\text{Tr}(\rho_{\mathcal{E}}^2)$ for ‘measuring’ of how well a quantum operation maintains the purity of an input state [27].
("entropy exchange", <code>qregisterρ</code> , <code>qoperation\mathcal{E}</code>)	Entropy exchange $S(\rho, \mathcal{E}) = S(W)$, where W is related to the operation elements E_i by $W_{ij} = \text{Tr}(E_i \rho E_j^\dagger)$. The entropy exchange is a measure for the information exchanged between a system in the state ρ and its environment during the evolution \mathcal{E} [35].
("coherent information", <code>qregisterρ</code> , <code>qoperation\mathcal{E}</code>)	Coherent information $I(\rho, \mathcal{E}) = S(\mathcal{E}(\rho)) - S(\rho, \mathcal{E})$, where $S(\rho, \mathcal{E})$ is the entropy exchange. The coherent information measures the degree to which quantum coherence is maintained by \mathcal{E} [28,33].
("entanglement fidelity", <code>qregisterρ</code> , <code>qoperation\mathcal{E}</code>)	Entanglement fidelity $F_e(\rho, \mathcal{E}) = \sum_i \text{Tr}(\rho E_i) ^2$, where E_i are the operation elements of \mathcal{E} [28,33].
("channel fidelity", <code>qoperation\mathcal{E}</code>)	The channel fidelity denotes a special case of the entanglement fidelity using a maximally entangled pure state $ \Phi\rangle$: $F_c(\mathcal{E}) = \langle \Phi (\mathcal{I} \otimes \mathcal{E})(\Phi) \Phi \rangle$ [34].
("average fidelity", <code>qoperation\mathcal{E}</code> , <code>qoperatorU</code>)	Average (gate) fidelity $\bar{F}(\mathcal{E}, U) = \frac{F_J(\mathcal{E}, U)^{d+1}}{d+1}$, where F_J stands for the Jamiołkowski fidelity and d is the dimension of the space on which \mathcal{E} and U act [27,36].
("phase-space contraction", <code>qoperation\mathcal{E}</code>)	Phase-space contraction as a consequence of the dissipation process $\mathcal{E}(\rho)$; this is essentially a measure of the channel’s nonunitarity given by $\eta = N \text{Tr}[(\mathcal{E}(\rho_I) - \rho_I)^2]$, where $N = \dim \mathcal{H}$ and $\rho_I = I/N$ is the maximally mixed state [38].

archive includes several example worksheets (see the next section) and a `Read.me` file to briefly explain the installation of the program package. After proper installation, the program can be loaded like any other module of MAPLE by using the command `with(Feynman)`.

4. Interactive work using the FEYNMAN procedures: Examples

To illustrate the interactive use of the FEYNMAN procedures, four short examples will be displayed below. They provide a first impression of how the FEYNMAN commands can be used for computations that are straightforward but tedious to perform otherwise. Of course, the same procedures can be utilized also for dealing with more complex tasks as indicated by our third example below.

In the following, we assume that the FEYNMAN program has been ‘loaded’ at the beginning of the session and that MAPLE prompt is available for interactive work. Moreover, since several procedures result in rather a large MAPLE output, a colon (instead of a semicolon) is used to terminate several of the input lines in order to suppress the corresponding printout to screen.

4.1. Local amplitude damping

In a first example, we demonstrate the effect of a channel that acts only locally on one subsystem of a two-qubit state. To this end, let us consider a two-qubit system in the $|11\rangle$ state (e.g., two excited ions in a trap). Assuming that spontaneous emission of a photon is the dominant source of noise, we use the amplitude damping channel to model this effect which asymptotically takes a qubit to its ‘ground state’ $|0\rangle$.

To demonstrate the effect of this amplitude damping channel by means of the FEYNMAN program, we may start with specifying the initial two-qubit ‘excited’ state and the two *local* amplitude damping channels by typing

```
assume(p>=0 and p<=1);
initial_state := Feynman_set_qregister("11");
damping1      := qoperation(2, "amplitude damping", p, [1]);
damping2      := qoperation(2, "amplitude damping", p, [2]);
```

and where each channel acts on just one of the two qubits. In these input lines, moreover, we have assumed that the damping parameter (probability) is restricted to the range $p = 0..1$, which is made explicit by the trailing tildes also in the output below. Here, the syntax `qoperation(n, "...", [k])` means that we wish to ‘distribute’ the pre-defined (single-qubit) amplitude damping channel over an n -qubit space but where it should act only upon qubit no. k .

Having defined the initial state and the amplitude damping model, we may convince ourselves quite easily that these local channels really affect only one of the qubits, while the other one remains untouched. This is achieved by applying each of these channels to the initial state:

```
final_state1 := Feynman_apply(damping1, initial_state):
final_state2 := Feynman_apply(damping2, initial_state):
```

Instead of showing the internal MAPLE output of these commands, however, it's more convenient to use the short printout as provided by the command `Feynman_print()`:

```
Feynman_print(final_state1);
Feynman_print(final_state2);

p~|01><01| + (1 - p~) |11><11|
p~|10><10| + (1 - p~) |11><11|
```

As one can see from the first output line, with probability p the initial state $|11\rangle$ decays to $|01\rangle$ and is left unchanged with probability $1 - p$. In other words, the channel acts only on the first qubit. Of course, a similar statement can be made for the second output line and the second qubit. In order to allow this 'noise model' also for both qubits simultaneously, we may combine the action of both channels by either composing the local *two-qubit* channels from above, i.e. $\mathcal{E}_{12} = \mathcal{E}_2 \circ \mathcal{E}_1$, or by computing the tensor product $\mathcal{E}_{12} = \mathcal{E}_1 \otimes \mathcal{E}_2$ of the corresponding *single-qubit* channels:

```
combined_damping1 := Feynman_operation_compose(damping2, damping1):
single_qubit      := operation("amplitude damping", p):
combined_damping2 := Feynman_product("Kronecker", single_qubit, single_qubit):
```

Following similar lines as above, we can show now that the decay of the system affects both qubits

```
final_state3 := Feynman_apply(combined_damping1, initial_state):
final_state4 := Feynman_apply(combined_damping2, initial_state):
Feynman_print(final_state3);
Feynman_print(final_state4);

p~2 |00><00| - p~ (-1 + p~) |01><01| - p~ (-1 + p~) |10><10| + (-1 + p~)2 |11><11|

p~2 |00><00| - p~ (-1 + p~) |01><01| - p~ (-1 + p~) |10><10| + (-1 + p~)2 |11><11|
```

and that the result is the same, independent of the mathematical construction of these *two-qubit* channels. Of course, this is no surprise since both channels are equivalent (which is seen quite easily by looking at the corresponding operation elements). More formally, the equivalence of the two-qubit channels `combined_damping1` and `combined_damping2` can be checked by calculating the distance (measure) between these quantum operations. For the generalization of the trace-distance [cf. Section 2.4], for instance, this is obtained by the command:

```
Feynman_measures("J distance", combined_damping1, combined_damping2);
0.
```

4.2. Entanglement-breaking channel

In our second example, we compare the effect of two quantum operations on the entanglement of some input state. If some quantum operation \mathcal{E} turns any arbitrary input state into a separable output state then \mathcal{E} is said to be *entanglement-breaking*. In this case the quantum operation's dual state must be separable, too. Therefore, it is sufficient to investigate the separability of the dual states.

We begin by defining the depolarization and phase-damping channels, together with their corresponding dual states.

```

channel1 := qoperation("depolarize", p):
channel2 := qoperation("phase damping", p):
dual_state1 := Feynman_qoperation_representation("dual state", channel1):
dual_state2 := Feynman_qoperation_representation("dual state", channel2):

```

If we assume here that the decoherence strength of the channels increases monotonically with the decay parameter p , where $p \approx 1 - e^{-\Gamma t}$ is often taken in order to denote an exponential decay of the (sub-)system, we may ask of whether a channel becomes entanglement-breaking for $p \rightarrow 1$, i.e. for $t \rightarrow \infty$. For this limit, let us test the separability of the dual states by means of the command `Feynman_is_separable()`. In particular, let us choose the positive partial transpose (PPT) criterion as a necessary (but not sufficient!) test for separability:

```

Feynman_is_separable("PPT", eval(dual_state1, p=1), [[1],[2]]);

true

Feynman_is_separable("PPT", eval(dual_state2, p=1), [[1],[2]]);

true

```

As seen from this output, both dual states are separable for $p \rightarrow 1$ and, hence, both channels will eventually destroy all entanglement, independent from the input state. Again, this result might not be surprising since we expect that the interaction with the environment will asymptotically wash out the internal correlations of a quantum system. However, a more detailed analysis using e.g. the concurrence measure reveals the qualitative difference between the two noise models. This is seen from the concurrence of the two dual states as displayed in Fig. 2.

```

temp1 := Feynman_measures("concurrence", dual_state1):
temp2 := Feynman_measures("concurrence", dual_state2):
plot([temp1, temp2], p=0..1);

```

While, for the phase damping channel, the entanglement of the dual state is destroyed only asymptotically in time, it vanishes for a depolarization channel for $p \approx 2/3$. That is, the depolarization channel becomes entanglement-breaking already at some finite time t . This behavior is sometimes also called ‘sudden death of entanglement’.

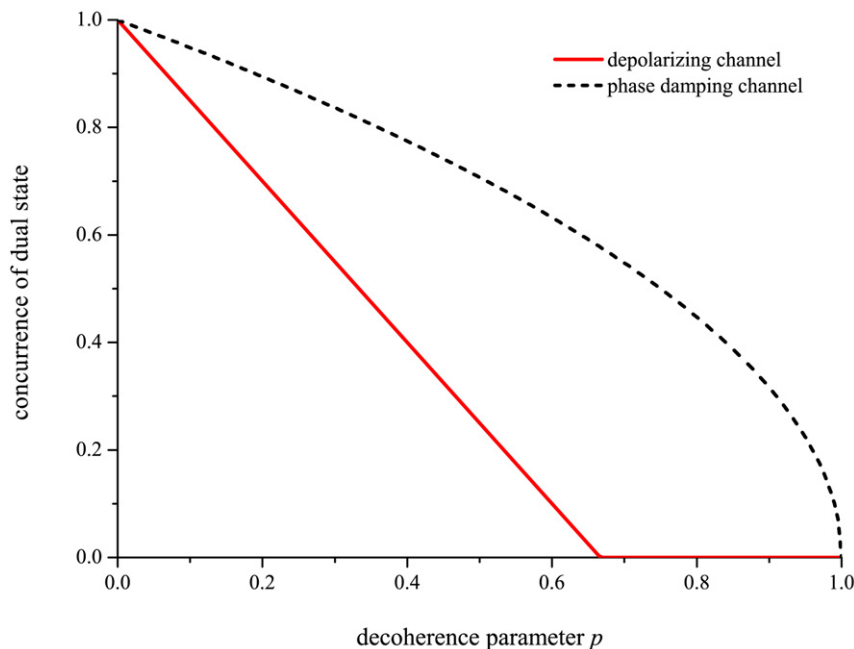


Fig. 2. Concurrence of the dual states as function of the decoherence parameter p . Results are shown for the depolarizing channel (solid) and the phase damping channel (dashed). While both channels are entanglement-breaking for $p \rightarrow 1$, the depolarizing channel becomes entanglement-breaking already at finite times, i.e. for $p < 1$.

4.3. Entanglement vs. purity: Characterization of single-qubit channels

In our third example, we attempt to characterize several single-qubit channels with regard to their entanglement and purity properties. Using the duality between channels and the (dual) states in Eq. (13), this task is closely related to the investigation of how local (single-qubit) channels affect a two-qubit Bell state, a case study that was done similarly by Ziman and Bužek [37].

To create an ‘entanglement-vs.-purity’ phase diagram for a few selected channels, let us investigate the entanglement and purity of the corresponding dual states. Beside of some known channels [cf. Table 3], here we shall consider especially the class of (single-qubit) unital channels, i.e. channels that obey the condition $\mathcal{E}(\mathbb{1}) = \mathbb{1}$. Of course, most of the popular single-qubit channels are unital, since they are special cases of the implemented Pauli channel, $\mathcal{E}(\rho) = \sum_i \sigma_i \rho \sigma_i^\dagger$, $i = 0..3$, which provides a suitable basis for this class. A well-known exception is the amplitude damping channel which describes a *dissipative*, nonunital process. In general, the nonunitality of a quantum operation can be measured by means of the ‘phase-space contraction’ (measure) [38], which has been implemented also in the FEYNMAN program (see Table 4).

We begin by defining the channels and their dual states

```
channel1 := qoperation("phase damping", p):
channel2 := qoperation("amplitude damping", p):
channel3 := qoperation("depolarize", p):
channel4 := qoperation("Pauli", a,b,c,d):

dual_state1 := Feynman_qoperation_representation("dual state", channel1):
dual_state2 := Feynman_qoperation_representation("dual state", channel2):
dual_state3 := Feynman_qoperation_representation("dual state", channel3):
dual_state4 := Feynman_qoperation_representation("dual state", channel4):
```

For the first three channels, which just depend on the single parameter $p = 0..1$, we can evaluate both, the concurrence as a measure for entanglement and the purity of the dual states. The purity is defined as the trace of the squared density matrix, $\text{Tr}(\rho^2)$, which, in turn, is equivalent to the reciprocal of the so-called ‘participation ratio’ within the FEYNMAN program. For these channels, therefore, the entanglement-vs.-purity graphs can be generated as parametric plots

```
plot1 := plot([1/Feynman_measures("participation ratio", dual_state1),
  Feynman_measures("concurrence", dual_state1), p=0..1],
  Purity=0..1, Concurrence=0..1):
plot2 := plot([1/Feynman_measures("participation ratio", dual_state2),
  Feynman_measures("concurrence", dual_state2), p=0..1]):
plot3 := plot([1/Feynman_measures("participation ratio", dual_state3),
  Feynman_measures("concurrence", dual_state3), p=0..1]):
```

The (fourth) Pauli channel, in contrast, depends on the four parameters a, b, c, d which should be ‘normalized’ to $a + b + c + d = 1$. In the entanglement-vs.-purity diagram, this parameter space corresponds to some area which we can visualize most easily by (randomly) generating a number of such Pauli channels. However, no attempt is made to obtain a truly *uniform* distribution of these ‘random’ unital channels.

```
data4 := table():
for i from 1 to 1000 do
  a := RandomTools[Generate](float(range=0..1, method=uniform));
  b := RandomTools[Generate](float(range=0..1, method=uniform));
  c := RandomTools[Generate](float(range=0..1, method=uniform));
  d := RandomTools[Generate](float(range=0..1, method=uniform));
  temp := a+b+c+d;
  a := a/temp;
  b := b/temp;
  c := c/temp;
  d := d/temp;
  data4[i] := [1/Feynman_measures("participation ratio", eval(dual_state4,
    {'a'=a, 'b'=b, 'c'=c, 'd'=d})), Feynman_measures("concurrence",
    eval(dual_state4, {'a'=a, 'b'=b, 'c'=c, 'd'=d}))];
end do:
```

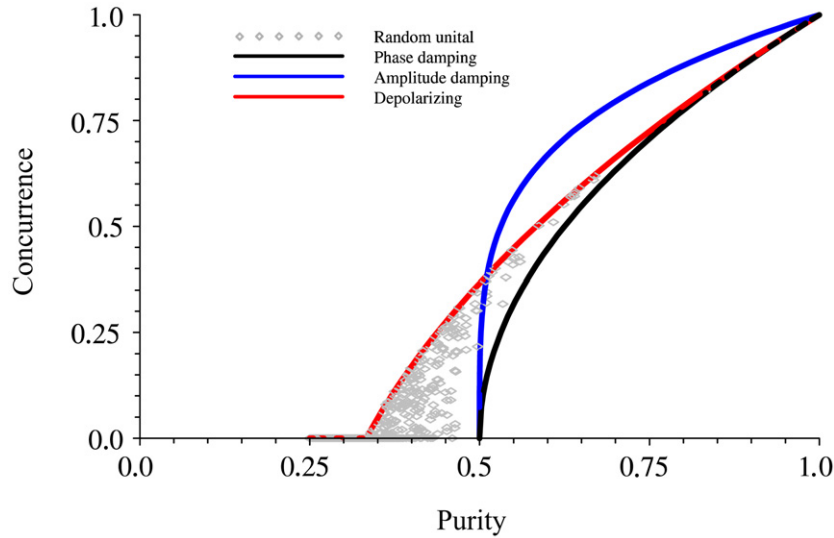


Fig. 3. Concurrence-vs.-purity phase diagram for a few selected single-qubit channels. The depolarization and phase damping channels mark the border of the area that is filled by the set of all unital channels. The amplitude damping is an example of a nonunital channel.

```
data4 := convert(data4, list):
plot4 := plot(data4, style=point):
```

Finally, we can display the four generated plot structures within a single figure

```
plots[display]([plot1, plot2, plot3, plot4]);
```

which is displayed below. As seen from Fig. 3, the entanglement and the amount of mixing in a quantum state tend to exclude each other, i.e. maximal entanglement is observed only for pure states [39]. In addition, we find that the depolarizing and phase damping channel coincide just with the borders of the area that is spanned by the class of all unital channels. As expected for a nonunital channel, the curve of the amplitude damping channel clearly leaves the ‘unital area’. In order to make this more explicit, we can use the above-mentioned phase-space contraction as a quantitative measure of a channel’s nonunitality.

```
Feynman_measures("phase-space contraction", qoperation("depolarize", p));
0.
Feynman_measures("phase-space contraction", qoperation("amplitude damping", p));
2
p~
```

4.4. Grover’s algorithm with noise

Our last example demonstrates the impact of different noise types on the result of a simple quantum algorithm. As an example we take Grover’s quantum algorithm for searching an unstructured database [5,6]. For the sake of simplicity, we consider the simple case of a four-element database which is represented by two-qubit quantum register. In our example, the desired search result will be given by the $|01\rangle$ basis state (which is just an arbitrary choice).

The circuit implementing the search algorithm is shown in Fig. 4. At the beginning, the two qubits representing the database are initialized in the $|00\rangle$ state and then brought into an equal superposition $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$ of all four basis states. This is achieved by applying a Hadamard gate to both of the qubits. Then the first essential step is carried out by applying the so-called ‘oracle’ which marks the searched state (in our example $|01\rangle$) by changing the sign of its coefficient so that the register is now in the state $\frac{1}{2}(|00\rangle - |01\rangle + |10\rangle + |11\rangle)$. This oracle operator corresponds to $I - |01\rangle\langle 01|$ and is implemented by the gate sequence 1–3. However, after the oracle query, a measurement would still not distinguish the searched state from the other states since the absolute values and hence the measuring probabilities are the same. Therefore, the second essential step in Grover’s algorithm is to transform the phase difference of the searched state into an amplitude difference. This is achieved by an ‘inversion about the

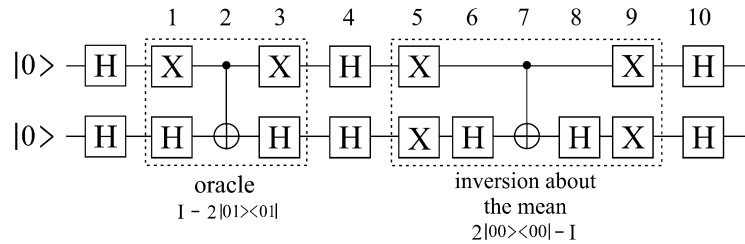


Fig. 4. Grover search algorithm for two qubits. See the main text for an explanation of the individual steps.

mean' operation which is implemented by the steps 5–9 in Fig. 4 (see, e.g., [6]). The key idea is that this step amplifies the searched state while suppressing the unwanted states. This is repeated a certain number of times so that, finally, only the correct result of the search problem is measured with a probability close to unity. In the general n -qubit case (corresponding to a search space with $N = 2^n$ database elements), the steps 1–10 have to be repeated $\approx \frac{\pi}{4}\sqrt{N}$ times (for $n \gg 1$). In our (ideal) example case, a single application of this procedure is already enough to transform the register to the state $|01\rangle$, i.e. a measurement will yield the correct search result with certainty.

Using the FEYNMAN program, we want to explore how imperfections and noise may effect the search algorithm and lead to deviations from the ideal result. To that aim, we will assume that between the idealized quantum gates a given quantum operation with noise strength p acts on the quantum register. For our demonstration, we simply use three of the predefined noise models: the 'fully correlated phase flip', the 'individual phase flip' and the 'fully correlated amplitude damping' model (see Table 3). In principle, any other (user-defined) model can be applied, but we note that with increasing number of qubits and increased complexity of the noise model the *algebraic* calculations become very slow. In these cases, it is helpful to restrict oneself to purely numerical calculations.

To model the noisy quantum search algorithm as described above, we begin by defining the initial state $|00\rangle$ and directly create the superposition of the basis states by applying two Hadamard gates.

```
> initial_state := Feynman_apply(qoperator("HH"), Feynman_set_qregister("00"));
                                     [1/2]
                                     [  ]
                                     [1/2]
initial_state := qregister("00", 2, [  ])
                                     [1/2]
                                     [  ]
                                     [1/2]
```

Now we define the steps 1–10 as given in Fig. 4.

```
> step[1] := qoperator("XH");
> step[2] := qoperator("cnot");
> step[3] := qoperator("XH");
> step[4] := qoperator("HH");
> step[5] := qoperator("XX");
> step[6] := qoperator("IH");
> step[7] := qoperator("cnot");
> step[8] := qoperator("IH");
> step[9] := qoperator("XX");
> step[10] := qoperator("HH");
```

The next commands define the three different (two-qubit) noise models A , B , C that will be applied between the quantum gates. Each of the models uses some symbolic parameter $0 \leq p \leq 1$ to control the decoherence strength.

```
> assume(p>=0 and p<=1);
> noise_A := qoperation("fully correlated phase flip", p, 2);
> noise_B := qoperation("individual phase flip", p, 2);
> noise_C := qoperation("fully correlated amplitude damping", p, 2);
```

Finally, we use a simple loop construction in order to successively apply the steps 1–10 and apply one of the three noise models A , B , C before each step, respectively. This results in three different final states, one for each noise model.

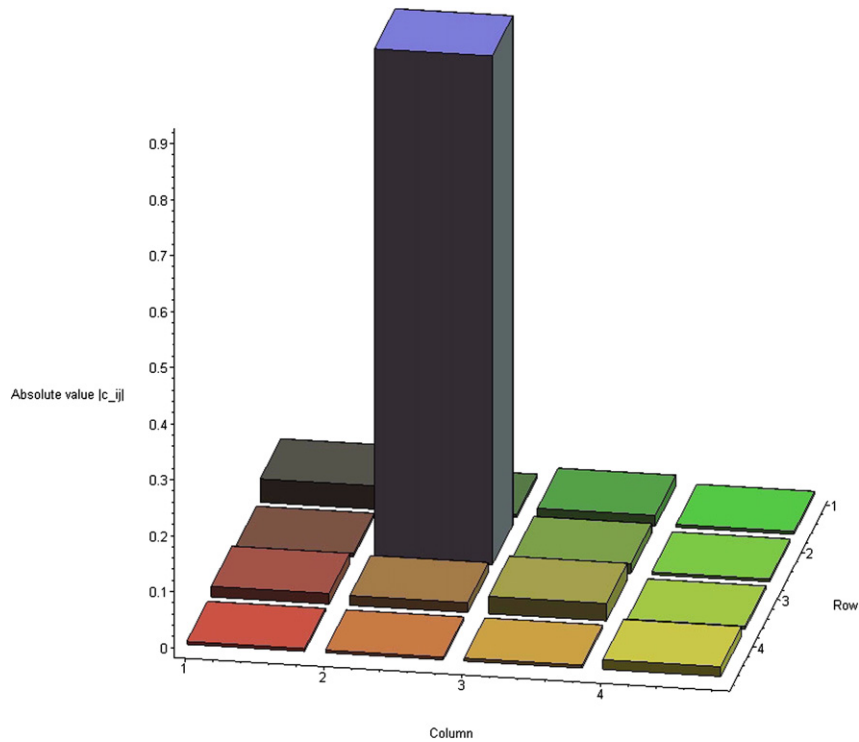


Fig. 5. Visualization of the final-state density matrix resulting from the noisy search algorithm where noise model C was applied with a strength parameter $p = 0.05$ after each quantum gate.

```

> temp_A := initial_state:
> temp_B := initial_state:
> temp_C := initial_state:
> for i from 1 to 10 do
    temp_A := Feynman_apply(noise_A, temp_A);
    temp_A := simplify(Feynman_apply(step[i], temp_A));
    temp_B := Feynman_apply(noise_B, temp_B);
    temp_B := simplify(Feynman_apply(step[i], temp_B));
    temp_C := Feynman_apply(noise_C, temp_C);
    temp_C := simplify(Feynman_apply(step[i], temp_C));
end do:
> final_state_A := temp_A:
> final_state_B := temp_B:
> final_state_C := temp_C:

```

To get an impression of the different noise models' effects on the final state, one can plot the absolute values of the density matrix entries as shown in Fig. 5.

```
Feynman_plot_probability(eval(final_state_C, p=0.05));
```

For example, from Fig. 5 one can see that the fully correlated amplitude damping (model C with $p = 0.05$) generally leads to nonzero matrix elements although the second diagonal element (which corresponds to the correct final state $|01\rangle$) still clearly dominates. Additionally, one can recognize the tendency to populate the 'ground state' $|00\rangle$ which is just the asymptotic state of the applied dissipative noise model C .

Finally, in order to compare the different noise models, we might be interested, for example, in the fidelity between the real (noisy) final state and the ideal result. This is achieved by the following commands and the resulting curves are shown in Fig. 6.

```

> ideal_result := eval(final_state_A, p=0):
data_A := []:
data_B := []:
data_C := []:

```

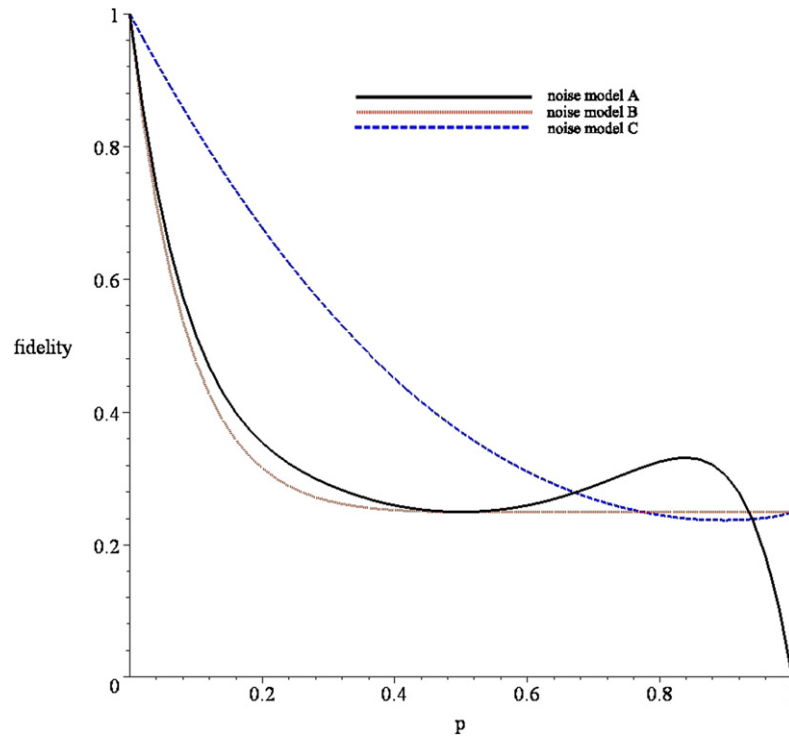


Fig. 6. Fidelity between the ideal final state of the quantum search algorithm and the noisy final states due to the models *A*, *B*, *C* with the noise parameter p as described in the text.

```

for i from 0 to 1 by 0.02 do
  data_A := [op(data_A), [i, Feynman_measures("fidelity", ideal_result,
    eval(final_state_A, p=i))]]:
  data_B := [op(data_B), [i, Feynman_measures("fidelity", ideal_result,
    eval(final_state_B, p=i))]]:
  data_C := [op(data_C), [i, Feynman_measures("fidelity", ideal_result,
    eval(final_state_C, p=i))]]:
end do:

> plot([data_A, data_B, data_C], 'p'=0..1, fidelity=0..1, legend=["noise model A",
  "noise model B", "noise model C"]);

```

5. Summary and outlook

Following our previous intention to provide a simple and flexible tool for dealing with n -qubit quantum systems, the current extension of the code now enables the user to manipulate quantum operations and, hence, to model the interaction of such systems with their environment. In particular, the command `Feynman_quantum_operation()` provides a very convenient access to a number of frequently applied ‘decoherence models’. Moreover, to support a quantitative analysis of the properties of the various channels, several measures for quantum operations and their dual states have been implemented by the procedure `Feynman_measures()`. The implementation of the Jamiołkowski dual states (via `Feynman_qoperation_representation()`) connects the data structure `qoperation()` for quantum operations to the `qregister()` structure for quantum registers, for which several tools were developed already in a previous version of the FEYNMAN program.

With the present extension, therefore, we made FEYNMAN ready for the simulation of decoherence in quantum registers, which is known today as one of the main obstacles in realizing various quantum information protocols. Apart from studying such ‘noise models’, however, an interesting further development concerns the implementation of active and/or passive error correction schemes. Using the command `Feynman_qoperation_invariants()`, already the current version can assist in finding the fixed points of a quantum operation, i.e. those matrix operators that are invariant under the action of the given channel. But further work is certainly needed and desirable in order to apply the FEYNMAN program in other fields of quantum information, such as error correction or use it for studying the entanglement in physical systems, e.g., the transfer of entanglement in atomic (photo-)ionization [40,41] or decay processes [42,43].

References

- [1] C.H. Bennett, G. Brassard, C. Crépeau, R. Josza, A. Peres, W.K. Wootters, *Phys. Rev. Lett.* 70 (1993) 1895.
- [2] A.K. Ekert, *Phys. Rev. Lett.* 67 (1991) 661.
- [3] C.H. Bennet, S.J. Wiesner, *Phys. Rev. Lett.* 69 (1992) 2881.
- [4] P.W. Shor, *SIAM J. Sci. Statist. Comput.* 26 (1997) 1484.
- [5] L.K. Grover, *Phys. Rev. Lett.* 79 (1997) 325.
- [6] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2000.
- [7] J. Preskill, *Lecture Notes on Quantum Information and Quantum Computation*, Physics, vol. 219, California Institute of Technology, 1998, <http://www.theory.caltech.edu/~preskill/ph229/>.
- [8] D. Boschi, S. Branca, F. De Martini, L. Hardy, S. Popescu, *Phys. Rev. Lett.* 80 (1998) 1121.
- [9] I.L. Chuang, N. Gershenfeld, M. Kubinec, *Phys. Rev. Lett.* 80 (1998) 3408.
- [10] L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, R. Cleve, I.L. Chuang, *Phys. Rev. Lett.* 85 (2000) 5452.
- [11] A.R. Calderbank, P.W. Shor, *Phys. Rev. A* 54 (1996) 1098.
- [12] D.A. Lidar, I.L. Chuang, K.B. Whaley, *Phys. Rev. Lett.* 81 (1998) 2594.
- [13] D. Kribs, R. Lafamme, D. Poulin, *Phys. Rev. Lett.* 94 (2005) 180501.
- [14] H.-P. Breuer, F. Petruccione, *The Theory of Open Quantum Systems*, Oxford University Press, 2002.
- [15] C.W. Gardiner, P. Zoller, *Quantum Noise*, third ed., Springer, Berlin, 2004.
- [16] T. Radtke, S. Fritzsche, *Comput. Phys. Comm.* 173 (2005) 91.
- [17] T. Radtke, S. Fritzsche, *Comput. Phys. Comm.* 175 (2006) 145.
- [18] M. Horodecki, P. Horodecki, R. Horodecki, *Phys. Lett. A* 223 (1996) 1;
A. Peres, *Phys. Rev. Lett.* 77 (1996) 1413.
- [19] K. Kraus, *States, Effects and Operations: Fundamental Notions of Quantum Theory*, Springer, 1983.
- [20] A. Jamiołkowski, *Rep. Math. Phys.* 3 (1972) 275.
- [21] P. Arrighi, C. Patricot, *Ann. Phys.* 311 (2004) 26.
- [22] K. Życzkowski, I. Bengtsson, *Open Sys. Inf. Dyn.* 11 (2004) 3.
- [23] W. Dür, M. Hein, J.I. Cirac, H.-J. Briegel, *Phys. Rev. A* 72 (2005) 052326.
- [24] R.A. Horn, C.R. Johnson, *Topics in Matrix Analysis*, Cambridge Univ. Press, New York, 1991.
- [25] K. Chen, L.A. Wu, *Quant. Inf. Comp.* 3 (2003) 193.
- [26] T. Havel, *J. Math. Phys.* 44 (2003) 534.
- [27] A. Gilchrist, N.K. Langford, M.A. Nielsen, *Phys. Rev. A* 71 (2005) 062310.
- [28] B. Schumacher, *Phys. Rev. A* 54 (1996) 2614.
- [29] B. Julia-Diaz, F. Tabakin, QDENSITY, <http://library.wolfram.com/infocenter/MathSource/5715/>;
P. Dumais, QUCALC, <http://library.wolfram.com/infocenter/MathSource/657/>;
T. Felbinger, QMATRIX, <http://library.wolfram.com/infocenter/MathSource/1893/>.
- [30] B. Butscher, H. Weimer, LIBQUANTUM, <http://www.enyo.de/libquantum/>;
B. Oemer, QCL, <http://tph.tuwien.ac.at/~oemer/qcl.html>;
Massively Parallel Ideal Quantum Computer Simulator (MPIQCS), <http://tph.tuwien.ac.at/~oemer/qcl.html>.
- [31] H. Nakazato, Y. Hida, K. Yuasa, B. Milotello, A. Napoli, A. Messina, *Phys. Rev. A* 74 (2006) 062113.
- [32] D.M. Tong, L.C. Kwak, C.H. Oh, J.L. Chen, L. Ma, *Phys. Rev. A* 69 (2004) 054102.
- [33] H. Barnum, M.A. Nielsen, B. Schumacher, *Phys. Rev. A* 57 (1998) 4153.
- [34] R. Jozsa, *J. Mod. Opt.* 41 (1994) 2315.
- [35] B. Schumacher, M.A. Nielsen, *Phys. Rev. A* 54 (1996) 2629.
- [36] M.A. Nielsen, *Phys. Lett. A* 303 (2002) 249.
- [37] M. Ziman, V. Bužek, *Phys. Rev. A* 72 (2005) 052325.
- [38] I. Garcia-Mata, M. Saraceno, M.E. Spina, G. Carlo, *Phys. Rev. A* 72 (2005) 062315.
- [39] D. Cavalcanti, F.G.S.L. Brandão, *Phys. Rev. A* 72 (2005) 040303(R).
- [40] T. Radtke, S. Fritzsche, A. Surzhykov, *Phys. Lett. A* 347 (2005) 73.
- [41] T. Radtke, S. Fritzsche, A. Surzhykov, *Phys. Rev. A* 74 (2006) 032709.
- [42] S. Fritzsche, *J. Electron Spectrosc. Relat. Phenom.* 114–16 (2001) 1155.
- [43] A. Surzhykov, P. Koval, S. Fritzsche, *Phys. Rev. A* 71 (2005) 022509.