Numerical strategies for quantum tomography: Alternatives to full optimization

Max S. Kaznady[*] and Daniel F. V. James[†]

Department of Physics and Centre of Quantum Information and Quantum Control, University of Toronto, 60 St. George Street, Toronto, Ontario, M5S 1A7, Canada

(Received 12 September 2008; published 9 February 2009)

We examine a variety of strategies for numerical quantum-state estimation from data of the sort commonly measured in experiments involving quantum-state tomography. We find that, in some important circumstances, an elaborate and time-consuming numerical optimization to obtain the optimum density matrix corresponding to a given data set is not necessary and that cruder, faster numerical techniques may well be sufficient; in other words, "the best" is the enemy of "good enough."

DOI: 10.1103/PhysRevA.79.022109 PACS number(s): 03.65.Wj, 03.67.−a

I. INTRODUCTION

The goal of quantum-state tomography [1–3] is to estimate, from a series of projective measurements performed on identically prepared quantum systems, the density matrix of the underlying ensemble of which these quantum systems are realizations. This process is necessarily nondeterministic in nature, relying on the frequency of experimental outcomes to estimate probabilities—a process that converges to the actual probabilities only in the infinite limit. Thus the reconstruction of the quantum state cannot be exact in any realistic experiment. Furthermore, these measurements can only yield estimates of the on-diagonal elements of the density matrix, but not directly any data about the off-diagonal elements. It is necessary to perform various unitary operations on the system (or, equivalently, to perform projective measurements in a variety of bases) in order to obtain such information about the complete state. Indeed, for a system with a discrete spectrum of n levels, the density matrix is specified by n^2-1 independent real parameters, and each parameter will require a separate measurement. Even after the required measurements have been performed, the experimenter faces the problem of estimating the density matrix from incomplete and noisy data. The problem is aggravated by the constraints that quantum physics places on the density matrix: It must be a non-negative, unit-trace Hermitian matrix. Today, the approach that is usually taken is to determine computationally what is the "best" such positive, unit-trace Hermitian matrix which corresponds to a particular data set and what confidence can we place on such an estimate. The most complicated such tomographic measurement performed to date [4], on an 8-qubit (256-state) system, realized in a trapped ion experiment, was *limited not by the experimental capabilities of the system, but rather by the complexity of the numerical-state recovery problem* [5]. This computational complexity, while underscoring the awesome computational potential inherent in quantum information, nevertheless presents an experimenter, intent on exploring larger and larger Hilbert spaces, with considerable tribulation when characterizing the performance of his or her apparatus.

In this paper we examine the problem from an entirely computational perspective. Specifically, we address the concern that maybe we are being too fastidious in approaching the state reconstruction problem. One can obtain a positive, unit-trace Hermitian matrix from tomographic data in a variety of ways. First, and most simply, one could generate a linear reconstruction of the noisy data (which tends to give a nonpositive matrix) and ensure positivity by setting the negative eigenvalues to zero, then renormalizing to ensure a unit trace. This we call the "quick and dirty" (QD) approach. A second strategy is to assume that the state must be nearly pure—after all, quantum technologies are usually in the business of trying to create pure states—and to simplify the computation by finding the pure state most compatible with the data. We call this the "forced purity" (FP) approach. A third approach is full optimization—i.e., the application of some constrained optimization routine—with a specific metric to define the "distance" between our data set and a positive density matrix, and search parameter space until the absolute best (i.e., global minimum) density matrix is obtained. Our goal is specifically to address the following questions: when is the rigorous optimization required, and when will some shortcut technique be good enough? This is a question that can only be addressed by simulation: since we need to know *a priori* the underlying density matrix of the ensemble to compare the recovered estimates. Starting with assumed density matrix, we employ a pseudorandom number generator to create some "pseudoexperimental data" with appropriate probability distribution. The various approaches to density matrix recovery are applied to it, and the result is compared with the initial density matrix to assess the accuracy of the recovery techniques. Our analysis concerns solely multiple correlated two-level systems—e.g., the qubits of a small-scale quantum computer; however, many of the techniques and results we present are readily adaptable to more general systems.

The paper is organized as follows: In Sec. II we discuss the generic tomography problem for a single qubit, which is generalized to the n-qubit case in Sec. III, describing specifically a number of memory management techniques required for scalability of the code, and our approach to the optimization routine (using gradient-based algorithms and employing the matrix differential calculus). The code itself is described in detail in Sec. IV and our results in Sec. V.

[*] max.kaznady@gmail.com

[†] dfvj@physics.utoronto.ca

II. ONE QUBIT

In this section, we will review the basic concept of quantum-state tomography by considering the estimation of a state of a single two-level system, or qubit.

A. Parametrizing the density matrix

The density operator describing the state of a system [6] is a Hermitian, non-negative definite operator of unit trace. The set of Pauli matrices [7] $\{\hat{\sigma}_0,\hat{\sigma}_1,\hat{\sigma}_2,\hat{\sigma}_3\}$ form, for a two-dimensional space, a complete orthonormal set of matrices, so that $\hat{\rho}$ can be expanded as a linear combination of $\hat{\sigma}_\mu$ as

$$\hat{\rho} = \sum_{\mu=0}^{3} r_\mu \hat{\sigma}_\mu, \tag{1}$$

where

$$r_\nu = \mathrm{Tr}\{\hat{\sigma}_\nu \hat{\rho}\}/2. \tag{2}$$

Since $\mathrm{Tr}\{\hat{\rho}\}=1$, $r_0=1/2$; further, since $\hat{\rho}^\dagger=\hat{\rho}$, the r_ν are all real parameters.

The r_ν may be determined experimentally as follows: Suppose we perform a measurement, specified by the projector $\hat{\Pi}_0$, on the system; the probability of obtaining a positive outcome is $\mathrm{Tr}\{\hat{\rho}\hat{\Pi}_0\}$. Repeating this measurement $\mathcal{N}$ times on identically prepared systems, the expected number of times we obtain this outcome will be

$$n_0 = \mathcal{N}\mathrm{Tr}\{\hat{\rho}\hat{\Pi}_0\} = \mathcal{N}\sum_{\mu=0}^{3}\mathrm{Tr}\{\hat{\sigma}_\mu\hat{\Pi}_0\}r_\mu. \tag{3}$$

If one repeated this procedure of multiple measurements for a set of four different measurement operators, $\{\hat{\Pi}_\nu\}$ $(\nu=0,1,2,3)$, one obtains a set of linear equations

$$n_\nu = \mathcal{N}\sum_{\mu=0}^{3} B_{\nu,\mu} r_\mu, \tag{4}$$

where

$$B_{\nu,\mu} = \mathrm{Tr}\{\hat{\sigma}_\mu \hat{\Pi}_\nu\}. \tag{5}$$

By choosing the measurement operators $\{\hat{\Pi}_\nu\}$ judiciously, one can ensure that $B_{\nu,\mu}$ is nonsingular, and hence that the desired parameters r_μ can be obtained from the observed quantities n_ν: viz.,

$$r_\nu = (\mathcal{N})^{-1}\sum_{\mu=0}^{3}(B^{-1})_{\nu,\mu} n_\mu. \tag{6}$$

Substituting r_ν into Eq. (1), we obtain the density matrix as a function of measurement outcomes, *provided the measurements have no noise or errors in them.*

Following the precedent of Ref. [8], we use the standard Stokes measurement basis for our numerical experiments. These measurement operators are given by

$$\hat{\Pi}_0 = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|), \quad \hat{\Pi}_1 = |0\rangle\langle 0|,$$

$$\hat{\Pi}_2 = |\bar{D}\rangle\langle\bar{D}|, \quad \hat{\Pi}_3 = |R\rangle\langle R|, \tag{7}$$

where $|0\rangle$ and $|1\rangle$ represent the two states of our qubits and

$$|R\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle), \tag{8}$$

$$|\bar{D}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{9}$$

A natural metric to compare the recovered density matrix $\hat{\rho}_{\mathrm{meas}}$ with the actual density matrix $\hat{\rho}_{\mathrm{true}}$ is the *fidelity* [9], defined as

$$F(\hat{\rho}_{\mathrm{meas}},\hat{\rho}_{\mathrm{true}}) = \{\mathrm{Tr}[(\sqrt{\hat{\rho}_{\mathrm{meas}}}\hat{\rho}_{\mathrm{true}}\sqrt{\hat{\rho}_{\mathrm{meas}}})^{1/2}]\}^2. \tag{10}$$

However, when we invert the measurement data linearly, our recovered “density matrix” $\hat{\rho}_{\mathrm{linear}}$ is not non-negative definite and hence we have the specific problem that fidelity turns out to be complex (not to mention the more general problem that $\hat{\rho}_{\mathrm{linear}}$ cannot be interpreted as a density matrix of a physical state). We have to correct the matrix obtained by linear reconstruction to obtain a proper density matrix.

B. Quick and dirty reconstruction

As a simple initial approach to this problem, we can decompose $\hat{\rho}_{\mathrm{linear}}$ into its spectral representation—i.e.,

$$\hat{\rho}_{\mathrm{linear}} = \hat{U}\hat{D}\hat{U}^\dagger, \tag{11}$$

where $\hat{D}$ is the diagonal matrix of eigenvalues (which are real, but not necessarily positive) and $\hat{U}$ is a unitary matrix. We then set all negative eigenvalues in $\hat{D}$ to zero, call this matrix $\hat{D}'$, and obtain

$$\hat{\rho}_{\mathrm{QD}} = \frac{\hat{U}\hat{D}'\hat{U}^\dagger}{\mathrm{Tr}\{\hat{D}'\}}.$$

This provides a rough initial estimate of the state; one of the goals of our analysis in this paper is to assess how good an estimate it is.

C. Forced purity

An alternative approach to the problem of obtaining a non-negative definite density matrix from measured data is to assume that the state is *pure*. Recall that for a pure state $|\Psi\rangle$ the density matrix can be described by a single ket as $\hat{\rho}_{\mathrm{pure}}=|\Psi\rangle\langle\Psi|$. Such a density matrix for n qubits has eigenvalue 0 with degeneracy 2^n-1 and eigenvalue 1 with degeneracy 1.

Because $\hat{\rho}_{\mathrm{pure}}$ is also Hermitian, it can be written in its spectral decomposition as

$$\hat{\rho}_{\mathrm{pure}} = \hat{V}\hat{D}_{\mathrm{pure}}\hat{V}^\dagger,$$

where $\hat{D}_{\mathrm{pure}}$ is the diagonal matrix with a single element equal to 1, all other elements being zero; $\hat{V}$ is a unitary matrix.

During linear inversion of a pure state, the eigenvalues of $\hat{\rho}_{\text{linear}}$ may be negative, but sufficiently close to eigenvalues of $\hat{\rho}_{\text{pure}}$. The idea of forcing purity on such a state is to obtain

$$\hat{\rho}_{\text{FP}} = \frac{\hat{V}_{\text{linear}}\hat{D}''\hat{V}_{\text{linear}}^{\dagger}}{\text{Tr}\{\hat{D}''\}},$$

where $\hat{D}''$ is the diagonal matrix obtained from $\hat{D}$ by setting the largest eigenvalue equal to 1, all others being equal to 0.

D. Maximum likelihood

Any Hermitian 2×2 non-negative unit-trace matrix can be uniquely parametrized using the Cholesky decomposition as

$$\hat{\rho}_{\text{ideal}}(t_1,t_2,t_3,t_4) = \frac{T^{\dagger}T}{\text{Tr}\{T^{\dagger}T\}}, \tag{12}$$

where

$$T(t_1,t_2,t_3,t_4) = \begin{pmatrix} t_1 & 0 \\ t_3 + it_4 & t_2 \end{pmatrix}. \tag{13}$$

Thus a "physical" density matrix can be specified by the four parameters $\vec{t}=\{t_1,t_2,t_3,t_4\}$. The ideal of the maximum-likelihood method is to perform a search of the $\vec{t}$ parameter space until we find a $\hat{\rho}_{\text{ideal}}(\vec{t})$ which is most likely to have generated the observed data $\{n_0,n_1,n_2,n_3\}$. To assess this likelihood, suppose that each datum n_{μ} is a statistically independent, Poisson-distributed random variable with expectation value $\bar{n}_{\mu}$. Further, if $\bar{n}_{\mu}$ is a large number, the Poisson distribution is well approximated by the Gaussian distribution—i.e.,

$$P(n_0,n_1,n_2,n_3) = \frac{1}{N_{\text{norm}}}\prod_{\nu=0}^{3} \exp\left[-\frac{(n_{\nu}-\bar{n}_{\nu})^2}{2\bar{n}_{\nu}}\right], \tag{14}$$

where N_{norm} is the normalization constant. If each datum n_{μ} is garnered from $\mathcal{N}$ repetitions of a measurement carried out on a system in state $\hat{\rho}_{\text{ideal}}(\vec{t})$, it is reasonable to make the identification $\bar{n}_{\nu}(t_1,t_2,t_3,t_4)=\mathcal{N}\langle\psi_{\nu}|\hat{\rho}_{\text{ideal}}(t_1,t_2,t_3,t_4)|\psi_{\nu}\rangle$, and the likelihood of a given parameter vector $\vec{t}$ generating the data $\{n_0,n_1,n_2,n_3\}$ can be obtained by substituting this identity into Eq. (14). We are then in a position to determine the parameter vector for which this probability is maximized, and hence the most likely density matrix. Instead of maximizing Eq. (14), it is equivalent, and mathematically more convenient, to minimize the following function:

$$\mathcal{L}(\vec{t}) = \frac{1}{2}\sum_{\nu=0}^{3}\frac{[n_{\nu}-\mathcal{N}\text{Tr}\{\hat{\Pi}_{\nu}\hat{\rho}_{\text{ideal}}(\vec{t})\}]^2}{\mathcal{N}\text{Tr}\{\hat{\Pi}_{\nu}\hat{\rho}_{\text{ideal}}(\vec{t})\}}. \tag{15}$$

In order to optimize this function efficiently, we need to compute its gradient. This is not an easy feat, as the closed analytic form does not simplify well and finite differencing is too inefficient. The situation becomes exponentially worse as we increase the number of qubits.

III. GENERALIZATION TO N-QUBITS

In the previous section, we outlined the possible routines for performing tomography of a single qubit. We now extend these routines to a higher number of qubits and see how the quick and dirty and forced purity methods compare to the elaborate and time-consuming maximum likelihood estimation (MLE) routine.

At first, the problem looks very simple—any state of each qubit is completely characterized by only four measurements. Hence, numerically the MLE procedure is rather easy to implement—we just need to optimize a function of four variables, which is achieved by the simplex or Powell optimization algorithm in a fairly short amount of time [10], without computing the gradient. However, 2 qubits, when correlated, are not characterized by 8 measurements, but by 4×4=16 measurements, because we are looking at a *system* of 2 qubits. If n is the number of qubits, then we would need to obtain 4^n measurement outcomes in some fixed 4^n-dimensional basis. Due to wave function collapse, we can only perform one projection measurement at a time (an outcome is an average over multiple identical projection measurements), and for each projection measurement on one qubit, we have to cycle through all possible combinations of projection measurements for the other qubits.

Let us introduce the following set of operators which generalize the Pauli matrices for n-qubit systems:

$$\hat{\Gamma}_{\mu} = \frac{1}{\sqrt{2^n}}\hat{\sigma}_{\mu_1}\otimes\hat{\sigma}_{\mu_2}\otimes\cdots\otimes\hat{\sigma}_{\mu_n}, \tag{16}$$

where $0\leqslant\mu_{\xi}\leqslant 3$ for all $1\leqslant\xi\leqslant n$ are the digits of the index μ in base 4. For example, if $\mu=33$ for a 4-qubit system, $\hat{\Gamma}_{33}=\hat{\sigma}_0\otimes\hat{\sigma}_2\otimes\hat{\sigma}_0\otimes\hat{\sigma}_1$, since 33 is equal to 0201 in base 4. For convenience, we have included a normalization constant, so that $\text{Tr}\{\hat{\Gamma}_{\mu}\hat{\Gamma}_{\nu}\}=\delta_{\mu,\nu}$ (in keeping with the convention used in Ref. [8]). Similarly, we write the projection operators for our measurement states as

$$\hat{\Pi}_{\nu} = \hat{\Pi}_{\nu_1}\otimes\hat{\Pi}_{\nu_2}\otimes\cdots\otimes\hat{\Pi}_{\nu_n}. \tag{17}$$

The Cholesky decomposition of $\hat{\rho}$ remains the same, except that $T(\vec{t})$ is a $2^n\times 2^n$ matrix specified by 4^n parameters $\vec{t}=\{t_1,t_2,\ldots t_{4^n}\}$; i.e.,

$$T(\vec{t}) = \begin{bmatrix} t_1 & 0 & 0 & 0 \\ t_{2^n+1}+it_{2^n+2} & t_2 & 0 & 0 \\ \vdots & & \ddots & \vdots \\ t_{4^n-1}+it_{4^n} & \cdots & t_{2^{n+1}-4}+it_{2^{n+1}-3} & t_{2^n} \end{bmatrix}. \tag{18}$$

A. Computational constraints and memory-efficient linear reconstruction

In order to perform computational numerical tomography in practice, we need to take the following into consideration.

(i) Computational efficiency: what is the upper bound on the number of floating point operations of a certain tomography algorithm?

TABLE I. Amount of memory required to store a $4^n \times 4^n$ complex floating-point matrix using 32 bits to store the real or imaginary part.

Qubits	Bytes	Gigabytes
1	1.28×10^2	1.28×10^{-7}
2	2.05×10^3	2.05×10^{-6}
3	3.28×10^4	3.28×10^{-5}
4	5.24×10^5	5.24×10^{-4}
5	8.39×10^6	8.39×10^{-3}
6	1.34×10^9	1.34
7	2.15×10^9	2.15
8	3.44×10^{10}	3.44×10^1
9	5.50×10^{11}	5.50×10^2
10	8.80×10^{12}	8.80×10^3
11	1.41×10^{14}	1.41×10^5

(ii) Amount of memory available: what is the upper bound on the size in computer memory of the largest data structure used by the tomography algorithm?

Kronecker tensor products increase the size of resultant matrices exponentially. The goal is to obtain the density matrix which has $2^n \times 2^n$ elements. So we cannot have any other data structure in memory which would be larger; otherwise, the problem of increasing the number of qubits becomes constrained by that particular data structure.

For example, consider the approach described in Ref. [8], in which a $4^n \times 4^n$ complex matrix $B_{\mu,\nu}$ [n-qubit generalization of the matrix defined by Eq. (5)] was stored in memory. Table I outlines how much memory is needed to store a $4^n \times 4^n$ complex floating point matrix using 32 bits to store the real or imaginary part.

It must also be noted that any type of storage media has to be able to perform read and write operations quite fast because this data structure would be accessed quite frequently. This is simply not the case for most conventional hard drives: Using standard personal computers of the type typically integrated into quantum optics laboratories, one is in practice limited to about 7 qubits, without resorting to more powerful computer hardware. *However, a data structure of maximum size of* $2^n \times 2^n$ *would allow to go as high as 15–16 qubits, at which point the density matrix itself would become a storage problem.* Thus our goal is to avoid storing matrix $B_{\mu,\nu}$ into memory. Instead, we can obtain its inverse element by element. This can be achieved as follows: The matrix $B_{\mu,\nu}$ for an n-qubit system is defined by the equation

$$\begin{aligned} B_{\nu,\mu} &= \mathrm{Tr}\{(\hat{\Pi}_{\nu_1} \otimes \cdots \otimes \hat{\Pi}_{\nu_n})(\hat{\sigma}_{\mu_1} \otimes \cdots \otimes \hat{\sigma}_{\mu_n})\} \\ &= \mathrm{Tr}\{\hat{\Pi}_{\nu_1}\hat{\sigma}_{\mu_1}\}\mathrm{Tr}\{\hat{\Pi}_{\nu_2}\hat{\sigma}_{\mu_2}\}\cdots\mathrm{Tr}\{\hat{\Pi}_{\nu_n}\hat{\sigma}_{\mu_n}\}. \end{aligned} \tag{19}$$

Defining the 4×4 matrix $\beta_{\nu_\xi,\mu_\xi} = \mathrm{Tr}\{\hat{\Pi}_{\nu_\xi}\hat{\sigma}_{\mu_\xi}\}$ for all $1 \leq \xi \leq n$, which can be easily inverted [provided a suitable set of measurements $\{\hat{\Pi}_\mu\}$ $(\mu = 0,1,2,3)$ has been chosen], we find

$$B^{-1}_{\nu,\mu} = \beta^{-1}_{\nu_1,\mu_1}\beta^{-1}_{\nu_2,\mu_2}\cdots\beta^{-1}_{\nu_n,\mu_n}, \tag{20}$$

where, as before, $(\nu_1, \nu_2, \ldots, \nu_n)$ are the base-4 digits of the index ν (and similarly for μ).

This allows us to calculate the initial linear reconstruction of the density matrix from the observed data $(n_0, n_1, \ldots, n_{4^n-1})$: viz,

$$\hat{\rho}_{\mathrm{linear}} = \sum_{\nu=0}^{4^n-1} \hat{\Gamma}_\nu r_\nu, \tag{21}$$

where

$$r_\nu = (\mathcal{N})^{-1} \sum_{\mu=0}^{4^n-1} (B^{-1})_{\nu,\mu} n_\mu, \tag{22}$$

in a computationally efficient manner. Now the only size constraint on linear reconstruction is the density matrix itself. Of course, storing the projection measurement matrices $\hat{\Pi}_\nu$ is also problematic—a quick solution is to generate these matrices when they become needed—one can store certain tensor combinations which make up $\hat{\Pi}_\nu$ into memory and only tensor on additional combinations to obtain the desired $\hat{\Pi}_\nu$.

B. Maximum likelihood

Extending the maximum-likelihood function (MLF) from Eq. (15) to n qubits, we obtain

$$\mathcal{L}(\vec{t}) = \frac{1}{2}\sum_{\nu=0}^{4^n-1} \frac{[\mathcal{N}\mathrm{Tr}\{\hat{\Pi}_\nu \hat{\rho}_{\mathrm{ideal}}(\vec{t})\} - n_\nu]^2}{n_\nu}, \tag{23}$$

where to simplify calculations we assumed that we can approximate variance by the measurement outcome average in the denominator. Minimizing this function becomes a severe computational problem. Most gradient-free optimization routines are rather slow and only work well for a low number of dimensions, whereas here we have a number of dimensions which grows exponentially with the number of qubits. We have to use a numerical routine that is more efficient; this usually involves calculating the gradient and/or the Jacobian. The finite-differencing approach is too slow for computing the gradient (a fact which we verified computationally) because evaluating the MLF is exponentially inefficient. Hence, we require an analytic closed form for the gradient and/or the Jacobian matrix.

It should also be noted that if the region of optimization is convex, we are looking at a nonlinear convex optimization problem, for which a number of algorithmic approaches should work. We decided to take the simplest approach possible: Optimize the MLF with built-in constraints using an algorithm which works on both convex and nonconvex sets. We reduce the computation time by deriving an analytic form for the gradient. An alternative approach is to derive a different MLF with an external set of constraints and launch another convex optimization algorithm similar to linear programming [11,12].

1. Initial algorithmic attempts

The following algorithms were considered to optimize $\mathcal{L}$ [10,13], mostly because they are available in libraries such as GNU Scientific Library (GSL) [14]:

(1) Simplex method.
(2) Powell's quadratically convergent method.
(3) Levenberg-Marquardt nonlinear least squares.
(4) Conjugate-gradient method.
(5) BFGS algorithm.

Routines 4 and 5 need to be able to perform gradient computation and line search in an efficient manner and have to converge to the desired minimum, even if started far away from it.

Let us start with the line search routines—the following algorithms can be implemented for a line search.

(1) Successive parabolic interpolation.
(2) Newton's method.
(3) Golden section search (GSS).

In order to pick one algorithm out of these three, we need to first know if the region of optimization is convex or not, and if so, then how closely does it resemble a quadratic function.

2. Matrix calculus derivation

Regardless of the method chosen for the line search in Sec. III B 1, we still need an efficient way of computing the gradient. As established earlier, finite differencing requires too many function evaluations and does not satisfy our computational efficiency constraints.

The goal of this section is to find the gradient of $\mathcal{L}$ in closed form. This procedure can then be extended to finding second-order partial derivatives for the Hessian matrix and differentiation with respect to a constant for the line search routine.

We begin with the gradient derivation. This reduces to finding

$$\frac{\partial\mathcal{L}}{\partial t_\nu} = \frac{\partial\mathcal{L}}{\partial T}\frac{\partial T}{\partial t_\nu}. \tag{24}$$

Using matrix calculus, it suffices to find $\frac{\partial\mathcal{L}}{\partial T}$, which would be a matrix of size $2^n \times 2^n$ in our case. Certain elements of this matrix would represent the values of $\frac{\partial\mathcal{L}}{\partial t_\nu}$:

$$\begin{aligned}\frac{\partial\mathcal{L}(\vec{t})}{\partial T} = {} & \frac{\mathcal{N}}{\mathrm{Tr}\{T^\dagger(\vec{t})T(\vec{t})\}^2}\sum_{\nu=0}^{4^n-1}\left[\frac{\mathcal{N}\mathrm{Tr}\{\hat{\Pi}_\nu\hat{\rho}_{\mathrm{ideal}}(\vec{t})\} - n_\nu}{n_\nu}\right] \\ & \times\left[\mathrm{Tr}\{T^\dagger(\vec{t})T(\vec{t})\}\frac{\partial\mathrm{Tr}\{\hat{\Pi}_\nu T^\dagger(\vec{t})T(\vec{t})\}}{\partial T}\right. \\ & \left. - \mathrm{Tr}\{\hat{\Pi}_\nu T^\dagger(\vec{t})T(\vec{t})\}\frac{\partial\mathrm{Tr}\{T^\dagger(\vec{t})T(\vec{t})\}}{\partial T}\right].\end{aligned} \tag{25}$$

Defining the real quantities

$$A(\vec{t}) = \mathrm{Tr}\{T^\dagger(\vec{t})T(\vec{t})\} \tag{26}$$

and

$$B_\nu(\vec{t}) = \mathrm{Tr}\{\hat{\Pi}_\nu T^\dagger(\vec{t})T(\vec{t})\}, \tag{27}$$

we find

$$\mathrm{Tr}\{\hat{\Pi}_\nu\hat{\rho}_{\mathrm{ideal}}(\vec{t})\} = \frac{B_\nu}{A}. \tag{28}$$

Further, we will denote the matrix derivatives of these quantities with respect to the Cholesky matrix T as follows:

$$B'_\nu(\vec{t}) = \frac{\partial\mathrm{Tr}\{\hat{\Pi}_\nu T^\dagger(\vec{t})T(\vec{t})\}}{\partial T}, \tag{29}$$

$$A'(\vec{t}) = \frac{\partial\mathrm{Tr}\{T^\dagger(\vec{t})T(\vec{t})\}}{\partial T}. \tag{30}$$

Because matrix calculus is only well defined for real-valued matrices, let us write

$$T = T(\vec{t}) = X + iY, \quad \hat{\Pi}_\nu = K_\nu + i\Lambda_\nu. \tag{31}$$

Then, using the matrix calculus theorems in Sec. III A, we find

$$A' = 2X + i2Y, \tag{32}$$

$$B'_\nu = 2XK_\nu - 2Y\Lambda_\nu + i(2X\Lambda_\nu + 2YK_\nu). \tag{33}$$

Denoting

$$C_\nu = \left[\frac{NB_\nu - An_\nu}{An_\nu}\right], \quad D_\nu = \left[\frac{AB'_\nu - B_\nu A'}{A^2}\right], \tag{34}$$

we find that the matrix derivative of $\mathcal{L}$ can be written in the compact form

$$\frac{\partial\mathcal{L}}{\partial T} = \mathcal{L}'(\vec{t}) = \mathcal{N}\sum_{\nu=0}^{4^n-1} C_\nu D_\nu, \tag{35}$$

where C_ν is a scalar and D_ν is a $2^n \times 2^n$ matrix. In fact, the upper diagonal of $\mathcal{L}'(\vec{t})$ and imaginary part of the diagonal are of no use to us—values of the gradient are seeded in the original locations of t_ν, so $\mathcal{L}'(\vec{t})$ has to be disassembled into real and imaginary parts and then the gradient vector has to be filled from the resulting matrices.

IV. DESCRIPTION OF THE CODE

The goal is to scale tomography routines up to a higher number of qubits on a standard single-processor workstation by refining the tomography algorithms to remove the numerical complexity bottleneck from experimental post-processing. In this section, we describe how the codes were implemented.

A. State tomography routine

Four routines provide tomography and run in the following order.

(1) Linear reconstruction—provides the linear reconstruction of the data by inverting the measurements into a matrix

$\hat{\rho}_{\text{linear}}$, outlined in Sec. III A, which has all the characteristics of a density matrix, except positive semidefiniteness.

(2) Quick and dirty—quickly fixes $\hat{\rho}_{\text{linear}}$ into $\hat{\rho}_{\text{QD}}$ by setting all negative eigenvalues to zero and renormalizing.

(3) Forced purity—for pure states, eigenvalues are known. This routine forces eigenvalues of $\hat{\rho}_{\text{linear}}$ or $\hat{\rho}_{\text{QD}}$ (does not matter which one) into those of a pure state, also ensuring a unit-trace condition.

(4) MLE—we use the elements of the quick and dirty density matrix as a starting point for our optimization routine. We then launch the BFGS2 algorithm supplied with GSL.

Our progress while developing these routines is as follows:

(1) Started with our own simplex method code in MATLAB, which only optimized 4 qubits—a gradient-based algorithm was needed.

(2) Wrote the conjugate-gradient routine in MATLAB using a GSS line search routine and using a finite-difference gradient, which allowed for 5–6-qubit tomography.

(3) Applied matrix differential calculus to the gradient and obtained a closed-form expression, which severely improved MATLAB routines for up to 7 qubits.

(4) Experimented with Newton's method and successive parabolic interpolation line searches, which did not work in the end. This led us to suspect that the region of optimization is not convex.

(5) Rewrote everything in C using GSL and employed GSL's BFGS2 algorithm and its collection of line searches—this pushed our routines to 9 qubits (MLE limits to 9 qubits, but not forced purity).

All codes are currently implemented in C using GSL, with prototype routines also available in MATLAB.

B. Creating pseudo-experimental data

Generally, if one wants to simulate a physical state characterized by $\hat{\rho}_{\text{physical}}$ with $100\epsilon\%$ experimental-state error, then

$$\hat{\rho}_{\text{physical}} = (1-\epsilon)\hat{\rho}_{\text{theoretical}} + \epsilon\hat{\rho}_{\text{random}},$$

where $\hat{\rho}_{\text{theoretical}}$ is a density matrix of some desired state and ϵ is a real-valued constant, which simulates experimental "state error"—the physical state always differs from the intended state by some small amount; a random density matrix is created as follows

$$R = 2\text{rand}(2^n) - 1 + i[2\text{rand}(2^n) - 1], \tag{36}$$

$$\hat{\rho}_{\text{random}} = \frac{R^{\dagger}R}{\text{Tr}\{R^{\dagger}R\}}, \tag{37}$$

where the rand function creates a $2^n \times 2^n$ matrix of pseudorandom values, sampled from a Uniform (0,1) distribution [15].

For instance, the following results in a noisy Greenbengen-Horne-Zeilinger (GHZ) state:

$$\hat{\rho}_{\text{GHZ}} = (1-\epsilon)\frac{1}{2}|100\ldots01\rangle\langle100\ldots01| + \epsilon\hat{\rho}_{\text{random}}.$$

The simulation routine creates a physical density matrix, simulates experimental measurement outcomes, and then attempts to reconstruct this density matrix. Knowing what the reconstructed density matrix should be enables us to compare how well each reconstruction routine works for a certain number of qubits.

The expected number of positive outcomes is obtained using Eq. (3): viz.,

$$\bar{n}_{\nu} = \mathcal{N}\text{Tr}\{\hat{\Pi}_{\nu_1} \otimes \hat{\Pi}_{\nu_2} \otimes \cdots \otimes \hat{\Pi}_{\nu_n}\hat{\rho}_{\text{physical}}\},$$

where $\mathcal{N}$ is a constant which is equivalent to the number of times repeated projective measurements were taken.[1] We then add experimental noise to the measurements[2] using

$$n_{\nu} = \text{Poisson}(\bar{n}_{\nu}),$$

where Poisson(λ) generates a random number from a Poisson distribution with mean λ using a probability integral transformation.

V. RESULTS

In this section, we discuss the conclusions to be drawn from the numerical trials described in the previous sections. In particular, we address the question posed in the title of this paper: Do we always need an expensive MLE routine to perform tomography or would quick and dirty or forced purity methods suffice?

We compare the quick and dirty and forced purity routines to the MLE routine for states with wide variations of entropy and entanglement. We also show how well these routines scale in run-time and how experimental errors affect the reconstructed states as the number of qubits increases.

The linear entropy, which specifies the degree of purity of the state, is defined as

$$S_{\text{linear}}(\hat{\rho}) = \frac{2^n}{2^n - 1}[1 - \text{Tr}\{\hat{\rho}\}]$$

for n qubits.

The tangle (i.e., the square of the concurrence [18]) is defined for 2 qubits as

$$\tau = [\max\{\lambda_4 - \lambda_1 - \lambda_2 - \lambda_3, 0\}]^2,$$

where λ's are the square roots of the eigenvalues of the matrix $\sqrt{\hat{\rho}}(\hat{\sigma}_y \otimes \hat{\sigma}_y)\hat{\rho}^*(\hat{\sigma}_y \otimes \hat{\sigma}_y)\sqrt{\hat{\rho}}$, which is guaranteed to be Hermitian [16], $\hat{\sigma}_y \otimes \hat{\sigma}_y$ is the spin-flip matrix, and $\hat{\rho}^*$ is the complex conjugate of density matrix $\hat{\rho}$. For larger numbers of qubits, it can be used as a lower bound on the degree of entanglement [17].

[1] $\mathcal{N}$ was set to 10^4 in our tomographic routine, but it can be any positive integer as long as n_{ν} values resemble realistic photon counts, and not fractions less than 1.

[2] Note that if the measurements are performed with zero noise, then the linear reconstruction routine performs tomography of the density matrix with fidelity value of 1.

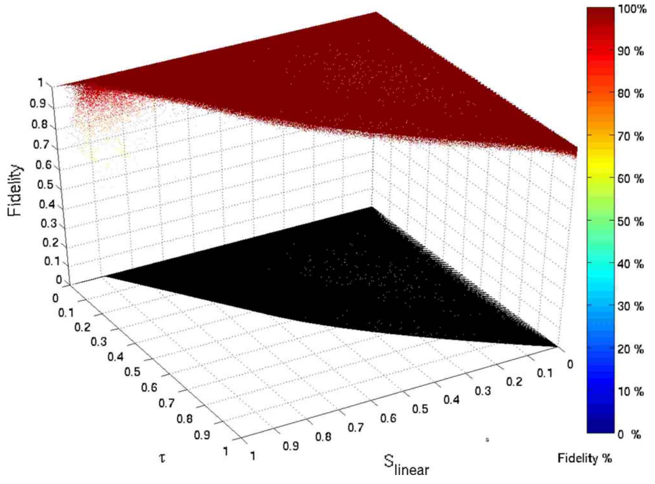

FIG. 1. (Color online) On the S_{linear} and τ plane one can see that the 2×10^6 generated states covered the entire plane; above each point is the corresponding fidelity of the recovered state using maximum likelihood. Notice that most fidelity values lie between 90% and 99%. The fidelity values create a thin plane, which suggests that the standard deviation is low for various states. However, some high-entropy states cannot be recovered well even with the expensive MLE procedure.

A. Linear entropy vs tangle plane

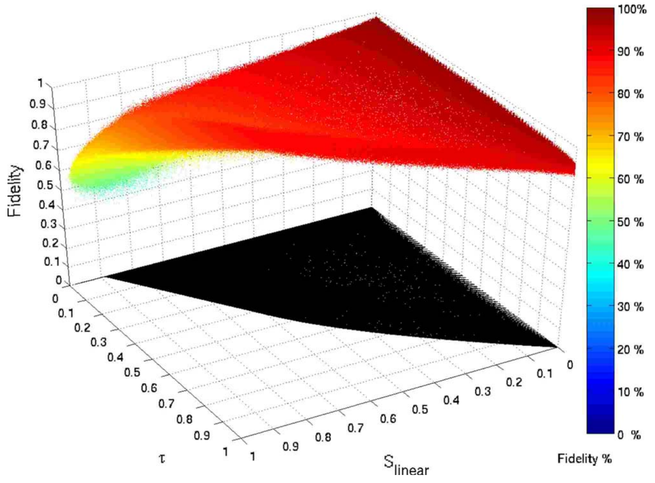

FIG. 2. (Color online) The same 2×10^6 states as in the previous figure and the projection on S_{linear} and τ plane is identical. Notice how the quick and dirty routine also fails to reach high fidelity values as the states become more mixed. However, for pure states quick and dirty is comparable to MLE in fidelity values.

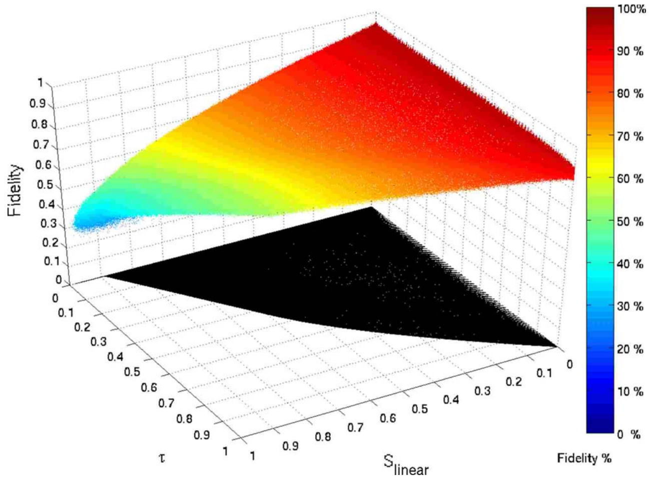

FIG. 3. (Color online) Again, same 2×10^6 states, but this time with forced purity performed. Notice that for 2 qubits forced purity appears to be worse than quick and dirty even for pure states—this is not the case when the number of qubits increases, as we explore in the next section.

We observed that for 2 qubits, certain states produce fidelities of over 90% using the quick and dirty routine and consistently high fidelities using MLE (see Figs. 1–3). We generated 2×10^6 pseudorandom density matrices, which filled the entire entropy-tangle plane. Random density matrices had to be biased in order to evenly cover the entire plane. For example, to fill the plane below the Werner state line, we used

$$\hat{\rho}_\tau = \begin{bmatrix} 1-\delta^2 & 0 & 0 & \delta\sqrt{1-\delta^2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \delta\sqrt{1-\delta^2} & 0 & 0 & \delta^2 \end{bmatrix}, \tag{38}$$

which biased the tangle in

$$\hat{\rho}_{\text{trial}} = \epsilon^2\hat{\rho}_{\text{random}} + (1-\epsilon^2)\hat{\rho}_\tau,$$

where

$$0 \leqslant \epsilon \leqslant 1, \quad 0 \leqslant \delta \leqslant 1/\sqrt{2}.$$

Varying δ changes tangle and varying ϵ changes entropy. We cycled through 100×100 different combinations of δ and ϵ and for each setting performed 100 trials, sampling $\hat{\rho}_{\text{random}}$ from Uniform $(-1,1)$ probability distribution for each trial. We can also move along the maximally entangled mixed state (MEMS) line by varying γ:

$$\hat{\rho}_{\text{MEMS}} = \begin{bmatrix} g(\gamma) & 0 & 0 & \gamma/2 \\ 0 & 1-2g(\gamma) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \gamma/2 & 0 & 0 & g(\gamma) \end{bmatrix},$$

where

$$g(\gamma) = \begin{cases} \gamma/2, & \gamma \geqslant 2/3, \\ 1/3, & \gamma < 2/3, \end{cases}$$

and

$$\hat{\rho}_{\text{trial}} = \epsilon^2\hat{\rho}_{\text{random}} + (1-\epsilon^2)\hat{\rho}_{\text{MEMS}}.$$

We further sampled 1000×1000 different settings of ϵ and γ to fill the area around the MEMS line: In this case increasing ϵ increases the distance from the MEMS line.

This suggests that for states with low entropies (pure states), the quick and dirty routine should work in theory.

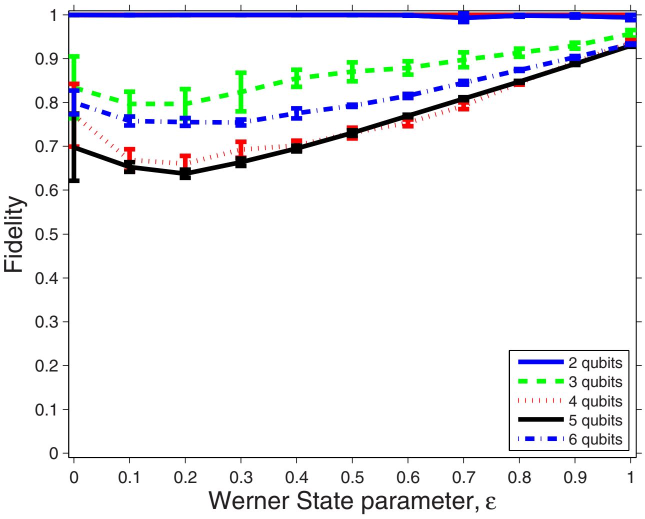

FIG. 4. (Color online) MLE works well for 2 qubits, but for a higher number of qubits there is a significant drop in fidelity. A 10% state error is quite large, but even at this error pure states are reconstructed better than mixed states as one can see around $\epsilon=1$ (highly entangled pure state).

This is not surprising, as from the spectral decomposition, we can see that all states with the $S_{\text{linear}}=0$ property, regardless of the value of τ, share one thing in common: eigenvalues. More precisely, for n qubits, eigenvalue 0 occurs with degeneracy 2^n-1 and eigenvalue 1 occurs with degeneracy 1. So setting negative eigenvalues to zero adjusts the eigenvalues closer to the eigenvalues of a pure state. If we know that the state is pure ahead of time, we can just reset the eigenvalues to the known values after the linear inversion procedure and obtain the density matrix—this is further explored in Sec. V C.

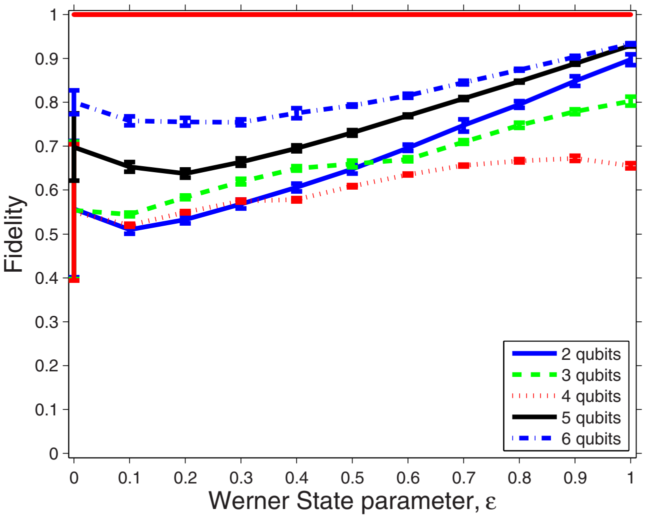

FIG. 5. (Color online) The quick and dirty routine does not perform very well even for pure states, although for a certain number of qubits it appears to work. On the contrary, it appears to improve as the number of qubits is increased.

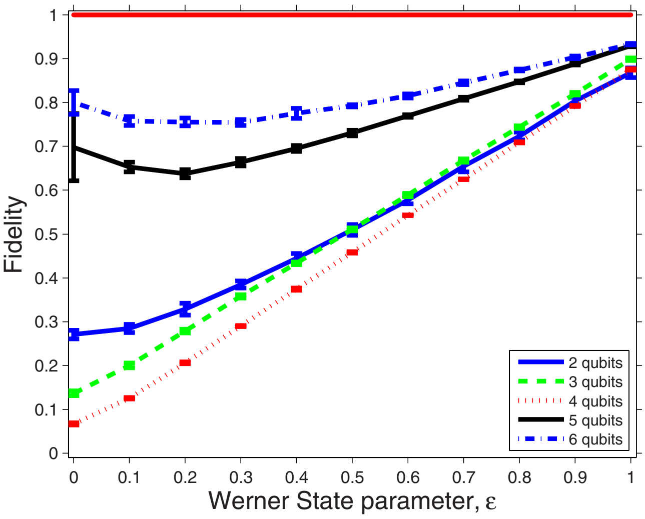

FIG. 6. (Color online) For "forced purity" we can confidently say that the routine improves significantly, as the number of qubits is increased. And for pure entangled states, this routine is almost as good as MLE, but runs in a fraction of the time.

B. Performance for *n* qubits

In order to extend this assessment to larger numbers of qubits, while still varying the amount of entanglement and disorder, we considered a generalized version of the Werner state for n qubits. Since this state slices through the entire plane presented in Sec. V A, we can see how well tomography operates on states with various tangle and entropy values by varying the location along the Werner-state line. An adjusted Werner-state density matrix is given by

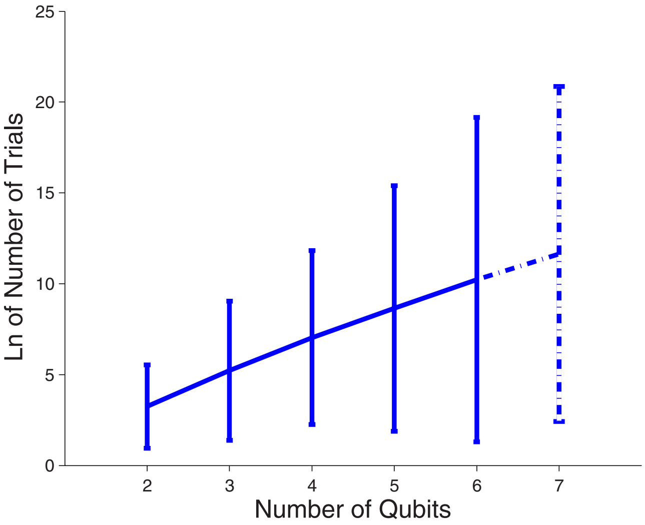

FIG. 7. (Color online) For each number of qubits, this plot shows the number of times each projection measurement has to be repeated for a state, in order to obtain an accurate estimate of the measurement outcome. With a 5% state error, this number of measurements will allow the forced purity routine to estimate the state with 90% fidelity. There is an exponential increase in how many times the experiment has to be repeated. Due to time constraints, only a few samples were obtained for 7 qubits.

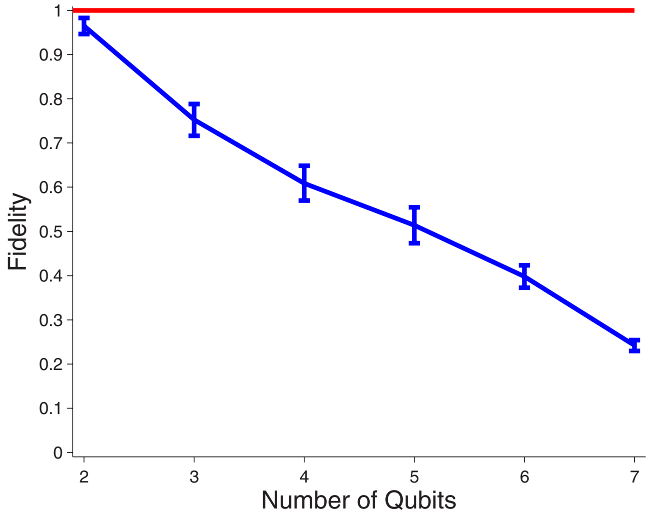

FIG. 8. (Color online) This graph shows the average fidelities of estimated pure states recovered using the quick and dirty approach for different numbers of qubits at 5% state error. In total 11 tangle values equally spaced between 0 and 1 were used and 10 recoveries performed for each tangle value. This shows that while quick and dirty appears to work for 2 qubits, in the long run it linearly worsens and cannot be used as a suitable tomography algorithm.

$$\hat{\rho}_{\text{Werner}} = |\text{GHZ}\rangle\langle\text{GHZ}|\epsilon + \frac{(1-\epsilon)}{2^n}\hat{I},$$

where

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}\{|00\dots0\rangle + |11\dots1\rangle\}$$

and $\hat{I}$ is the $2^n \times 2^n$ identity operator, representing a maximally mixed state. When $\epsilon=1$ we obtain a state located at $S_{\text{linear}}=0$ and $\tau=1$, and when $\epsilon=0$ we obtain $S_{\text{linear}}=1$ and $\tau=0$. We then vary ϵ from 0 to 1 in 101 increments and for each value of ϵ perform tomography 100 times, for a fixed number of qubits, with a 10% state error (see Figs. 4–6).

C. Forced purity tomography for pure states

In order for forced purity to work, the measurement outcomes have to be sufficiently close to their true values. To address the issue of "how close," we simulated pure states, with 11 distinct tangle values evenly spaced between 0 and 1, and then started with $\mathcal{N}=10$—i.e., $\mathcal{N}$ repeated projective measurements for each measurement outcome. We then increased $\mathcal{N}$ by 1 at each iteration and repeated forced purity tomography for some number of qubits. As soon as $\mathcal{N}$ allowed forced purity to perform tomography at 90% fidelity, the routine was terminated and the value for $\mathcal{N}$ recorded; see Fig. 7.

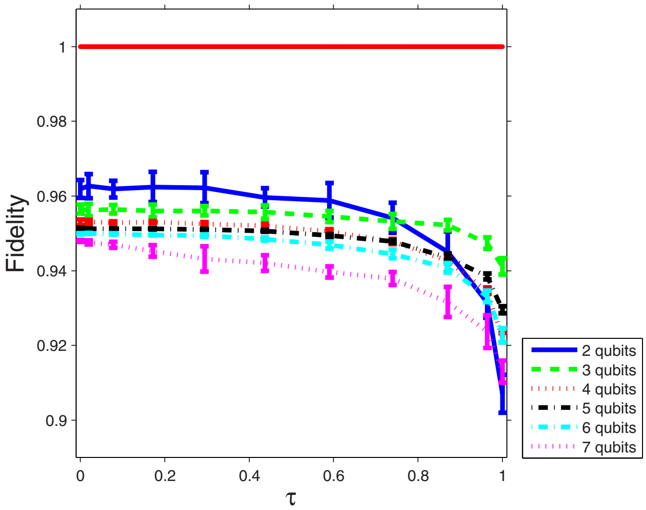

FIG. 9. (Color online) Using the same procedure as in Fig. 8, we then performed forced purity on each state. We said that for the Werner-state line the forced purity routine improves overall as the number of qubits increases. This is not the case for pure states, and as we can clearly see, forced purity slightly degrades, but overall still remains at over 90% fidelity even for 7 qubits.

One of the possible reasons why MLE tomography did not yield high fidelity values for a larger number of qubits is that it also required more accurate estimates of the measurement outcomes. Because MLE produced almost equal fidelities to forced purity for pure states, we would expect the same number of $\mathcal{N}$ to work for MLE tomography.

D. Run time and fidelity analysis

Section V C suggests that pure states do not require expensive MLE techniques for tomography. Nonetheless, it is interesting to see how MLE scales in run time compared to quick and dirty and forced purity routines. Here we present run times and fidelity estimates for a pure state with $\tau \approx 0.5$ and a slightly mixed state with the same tangle value which

TABLE II. Run times in milliseconds for a state at $\tau \approx 0.5$ and $S_{\text{linear}}=0$. "Iteration time" indicates how long the line search routine runs for each iteration of the BFGS routine. Abbreviations: MLE: complete maximum-likelihood reconstruction. QD: quick and dirty method. FP: forced purity.

n	MLE	Iteration time	QD	FP
2	$(4.0\pm1.0)\times10^{1}$	1.0 ± 0.5	$(1.9\pm0.5)\times10^{-1}$	$(1.7\pm0.4)\times10^{-1}$
3	$(1.0\pm0.3)\times10^{3}$	$(1.2\pm0.8)\times10^{1}$	$(2.9\pm0.3)\times10^{-1}$	$(2.6\pm0.2)\times10^{-1}$
4	$(1.6\pm0.3)\times10^{4}$	$(1.8\pm0.7)\times10^{2}$	1.1 ± 0.1	1.1 ± 0.2
5	$(3.5\pm0.6)\times10^{5}$	$(0.4\pm0.1)\times10^{4}$	9.4 ± 4.0	$(1.2\pm0.4)\times10^{1}$
6	$(6.0\pm0.3)\times10^{6}$	$(6.0\pm1.6)\times10^{4}$	$(5.8\pm0.3)\times10^{1}$	$(6.0\pm0.5)\times10^{1}$

TABLE III. Run times in milliseconds for a state at $\tau\approx 0.5$ and S_{linear} to that along the Werner state line; abbreviations are the same as in Table II.

n	MLE	Iteration time	QD	FP
2	$(4.1\pm1.0)\times10^{1}$	$(7.7\pm4.0)\times10^{-1}$	$(6.0\pm0.5)\times10^{-2}$	$(3.8\pm0.6)\times10^{-2}$
3	$(1.2\pm0.2)\times10^{3}$	$(1.2\pm2.0)\times10^{1}$	$(17\pm0.3)\times10^{-2}$	$(3.6\pm6.0)\times10^{-1}$
4	$(1.9\pm0.1)\times10^{4}$	$(2.0\pm1.0)\times10^{2}$	1.3 ± 0.4	1.1 ± 0.1
5	$(2.9\pm0.2)\times10^{5}$	$(2.9\pm0.9)\times10^{3}$	7.6 ± 0.6	8.8 ± 2.0
6	$(6.7\pm0.4)\times10^{6}$	$(7.0\pm2.0)\times10^{4}$	$(6.5\pm0.5)\times10^{1}$	$(6.7\pm0.5)\times10^{1}$

lies on the Werner-state line (see Tables II and III). We also show that even the expensive MLE routine decreases in fidelity as we increase the number of qubits. For this analysis we assume that an experiment can be performed a sufficiently large number of times, 10^6 to be exact.

In conclusion, forced purity results in lower fidelity values for 2 qubits than quick and dirty, but then increases in fidelity and converges to MLE's fidelity for a higher number of qubits (see Figs. 8 and 9).

VI. CONCLUSION

We have demonstrated that if the experiments can be performed a sufficient number of times, then using the forced purity routine, tomography can be performed in a quick and robust manner. However, as the entropy of a state increases, a much more expensive MLE routine has to be used to perform tomography, which does not scale well as the number of qubits increases. Quantum computing requires only pure-state tomography, for which we have obtained a scalable and efficient routine.[3]

ACKNOWLEDGMENTS

The authors would like to thank Robin Blume-Kohout, René Stock, and Rob Adamson for stimulating discussions and useful comments. This work was supported by the U.S. Army Research Office, NSERC, and Project OpenSource. We thank all the users of MATLAB File Exchange whose codes have been used to improve the plots in this paper

APPENDIX: MATRIX DIFFERENTIAL CALCULUS THEOREMS

The following theorems were used to derive an analytic expression to the gradient of the MLE function, which is more computationally efficient than the finite-difference gradient computation as the MLE function itself is expensive to evaluate.

If M is a real-valued matrix and $T=X+iY$, $T\in\mathcal{C}$, then [18–20]

$$\frac{\partial M}{\partial T}=\frac{\partial M}{\partial X}+i\frac{\partial M}{\partial Y}. \tag{A1}$$

The following are defined for real square matrices [20]:

$$\frac{\partial \mathrm{Tr}\{X^TY\}}{\partial X}=Y, \tag{A2}$$

$$\frac{\partial \mathrm{Tr}\{X^TX\}}{\partial X}=2X, \tag{A3}$$

$$\frac{\partial \mathrm{Tr}\{KX^TY\}}{\partial X}=YK, \tag{A4}$$

$$\frac{\partial \mathrm{Tr}\{KX^TX\}}{\partial X}=XK+XK^T. \tag{A5}$$

We begin by observing that

$$\mathrm{Tr}\{T^\dagger(\vec{t})T(\vec{t})\}=\sum_{\nu=0}^{4^n-1}t_\nu^2, \tag{A6}$$

which immediately implies that

$$\frac{\partial \mathrm{Tr}\{T^\dagger(\vec{t})T(\vec{t})\}}{\partial t_\nu}=2t_\nu.$$

Hence, using matrix calculus, we have the result

$$\frac{\partial \mathrm{Tr}\{T^\dagger(\vec{t})T(\vec{t})\}}{\partial T}=2X+i2Y. \tag{A7}$$

Equation (A7) is a compact means of stating the result of Eq. (A6) using a $2^n\times2^n$ matrix, where the value of the derivative is stored in the original position of t_ν in the Cholesky-decomposed matrix $T(\vec{t})$. This is the general idea behind *all* matrix calculus results we have used. We could have also obtained the same result by applying matrix calculus directly. For example, denote

$$\Phi=T^\dagger(\vec{t})T(\vec{t}).$$

Then, using Eqs. (A2) and (A3),

$$\frac{\partial \mathrm{Tr}\{\Phi\}}{\partial X}=2X+iY-iY=2X$$

and

[3] The run times of the routines mentioned in this paper can be improved linearly using parallel computation. However, because the complexity increases exponentially with the number of qubits, an efficient routine to performing tomography is crucial.

$$\frac{\partial \mathrm{Tr}\{\Phi\}}{\partial Y} = iX - iX + 2Y = 2Y.$$

This is consistent with our result in Eq. (A7).

Next, we set out to compute $\frac{\partial \mathrm{Tr}\{\hat{\Pi}_\nu \Phi\}}{\partial T}$. Recall that $\hat{\Pi}_\nu = K_\nu + i\Lambda_\nu$; hence,

$$\begin{aligned}\hat{\Pi}_\nu \Phi &= (K_\nu + i\Lambda_\nu)(X^TX + iX^TY - iY^TX + Y^TY)\\ &= K_\nu X^TX + K_\nu Y^TY + \Lambda_\nu Y^TX - \Lambda_\nu X^TY + i(K_\nu X^TY\\ &\quad + \Lambda_\nu X^TX + \Lambda_\nu Y^TY - K_\nu Y^TX). \qquad \text{(A8)}\end{aligned}$$

Applying Eqs. (A4) and (A5) to the real part of Eq. (A8), we obtain

$$\frac{\partial \mathrm{Tr}\{\hat{\Pi}_\nu \Phi\}}{\partial X} = XK_\nu + XK_\nu^T - Y\Lambda_\nu + Y\Lambda_\nu^T \qquad \text{(A9)}$$

and

$$\frac{\partial \mathrm{Tr}\{\hat{\Pi}_\nu \Phi\}}{\partial Y} = X\Lambda_\nu - X\Lambda_\nu^T + YK_\nu + YK_\nu^T. \qquad \text{(A10)}$$

Observe that $\forall \nu$, $\hat{\Pi}_\nu = \hat{\Pi}_\nu^\dagger$ yields $\Lambda_\nu = -\Lambda_\nu^T$ and $K_\nu = K_\nu^T$; therefore,

$$\frac{\partial \mathrm{Tr}\{\hat{\Pi}_\nu \Phi\}}{\partial X} = 2XK_\nu - 2Y\Lambda_\nu, \qquad \text{(A11)}$$

$$\frac{\partial \mathrm{Tr}\{\hat{\Pi}_\nu \Phi\}}{\partial Y} = 2X\Lambda_\nu + 2YK_\nu. \qquad \text{(A12)}$$

Substituting the above two equations into Eq. (A1), we obtain the result described in Sec. III B 2.

[1] U. Leonhardt, *Measuring the Quantum State of Light* (Cambridge University Press, Cambridge, England, 1997).

[2] Edited by M. G. A. Paris and J. Řeháček, *Quantum State Estimation, Lecture Notes in Physics*, Vol. 649 (Springer, Heidelberg, 2004).

[3] *Asymptotic Theory of Quantum Statistical Inference: Selected Papers*, edited by M. Hayashi (World Scientific, Singapore, 2005).

[4] H. Häffner, W. Hansel, C. F. Roos, J. Benhelm, D. Chek-al-kar, M. Chwalla, T. Körber, U. D. Rapol, M. Riebe, P. O. Schmidt, C. Becher, O. Gühne, E. Dür, and R. Blatt, Nature (London) **438**, 643 (2005).

[5] H. Häffner (private communication).

[6] See, for example, C. Cohen-Tannoudji, B. Diu, and F. Laloë, *Quantum Mechanics* (Wiley, New York, 1977), Complement DIII.

[7] We assume the standard form for the Pauli matrices, with $\mathrm{Tr}\{\hat{\sigma}_\mu \hat{\sigma}_\nu\} = 2\delta_{\mu,\nu}$ $(\mu,\nu = 0,1,2,3)$; see, for example, C. Cohen-Tannoudji *et al.*, *Quantum Mechanics* [8], Complement AIV. The matrix $\hat{\sigma}_0$ is the two-dimensional identity operator.

[8] D. F. V. James, P. G. Kwiat, W. J. Munro, and A. G. White, Phys. Rev. A **64**, 052312 (2001).

[9] R. Jozsa, J. Mod. Opt. **41**, 2315 (1994).

[10] W. H. Press, *Numerical recipes in C: The art of scientific computing*, 2nd ed. (Cambridge University Press, Cambridge, England, 1992).

[11] R. L. Kosut, A. Shabani, and D. A. Lidar, Phys. Rev. Lett. **100**, 020502 (2008).

[12] R. L. Kosut, I. Walmsley, and H. Rabitz, e-print arXiv:quant-ph/0411093.

[13] Michael T. Heath, *Scientific Computing: An Introductory Survey*, 2nd ed. (McGraw Hill, Toronto, 2002).

[14] GNU Scientific Library, http://www.gnu.org/software/gsl

[15] See the MATLAB online manual at http://www.mathworks.com

[16] W. K. Wootters, Phys. Rev. Lett. **80**, 2245 (1998).

[17] A. Wong and N. Christensen, Phys. Rev. A **63**, 044301 (2001).

[18] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Economics* (Wiley, Toronto, 1988).

[19] D. A. Turkington, *Matrix Calculus & Zero-One Matrices* (Cambridge University Press, New York, 2002).

[20] A. Graham, *Kronecker Products and Matrix Calculus: With Applications* (Halsted Press, Toronto, 1981).