Quantum Algorithm for Molecular Properties and Geometry Optimization

Ivan Kassal and Alán Aspuru-Guzik*
Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA 02138

(Dated: July 22, 2009)

It is known that quantum computers, if available, would allow an exponential decrease in the computational cost of quantum simulations. We extend this result to show that the computation of molecular properties (energy derivatives) could also be sped up using quantum computers. We provide a quantum algorithm for the numerical evaluation of molecular properties, whose time cost is a constant multiple of the time needed to compute the molecular energy, regardless of the size of the system. Molecular properties thus computed could also be used for geometry or other optimization, and we discuss the benefits of quantum techniques for Newton's method and Householder methods. Finally, global minima can be found using the quantum basin hopper algorithm, which offers an additional quadratic reduction in cost over classical multi-start techniques.

Applying *ab initio* methods of quantum chemistry to particular problems often requires computing derivatives of the molecular energy. For instance, a correct interpretation of spectroscopic experiments relies on the ability to compute derivatives with respect to nuclear coordinates and external electromagnetic fields. Likewise, computing the molecular gradient is the most commonly used method for the proper characterization of potential energy surfaces and for optimizing the geometry of all but the smallest molecules. The computation of these derivatives, known as *molecular properties*, is nowadays a routine matter when it comes to low-order derivatives or small systems (or both). This is largely due to advances in analytical gradient techniques, which allow for explicit property evaluation without resorting to numerical differentiation [1–6].

Nevertheless, the computation of higher-order derivatives is often prohibitively expensive, even though such derivatives are often needed. For example, third- and fourth-order anharmonic constants are usually required to accurately compute an absorbtion spectrum [3] or efficiently determine the location of transition states on complex potential energy surfaces [6]. Other properties of interest, such as hyperpolarizabilites, Raman intensities, or the vibrational circular dichroism, are also cubic or quartic derivatives. In this report, we show that quantum computers, once available, will be able to bypass some of the high cost of computing these properties. In particular, we show that any molecular property can be evaluated on a quantum computer using time resources that are independent of the size of the system under consideration, given a black box that can compute the molecular energy. That is, the evaluation of any property is, up to a small constant which is independent of the size of the system, as hard as computing the molecular energy once. We have previously characterized the exponential advantage of quantum computers at both computing molecular energies [7, 8] and simulating chemical reaction dynamics [9], and the present work extends that program to molecular properties.

We begin our paper with a brief overview of classical techniques of molecular property calculation, both numerical and analytical. We then introduce the quantum algorithm for molecular properties, and discuss its advantages and disadvantages with respect to classical techniques. We conclude with geometry optimization as a particular example, and we show that it can benefit from an additional quadratic speed-up through Grover's search [10].

THE CLASSICAL METHODS

Given an external perturbation $\boldsymbol{\mu}$, the total molecular electronic energy can be expanded in a Taylor series

$$E(\boldsymbol{\mu}) = E^{(0)} + \boldsymbol{\mu}^\top \mathbf{E}^{(1)} + \frac{1}{2}\boldsymbol{\mu}^\top \mathbf{E}^{(2)}\boldsymbol{\mu} + \dots \qquad (1)$$

where the coefficients $\mathbf{E}^{(n)}$ are called the *molecular properties* and describe the response of the system to the applied perturbation [11]. We consider time-independent properties, which can be obtained by differentiating the energy at $\boldsymbol{\mu} = \mathbf{0}$,

$$\mathbf{E}^{(n)} = \left.\frac{\mathrm{d}^n E}{\mathrm{d}\boldsymbol{\mu}^n}\right|_{\mathbf{0}}. \qquad (2)$$

Many examples of useful derivatives can be given. For instance, the derivatives with respect to the electric field $\mathbf{F}$ are the permanent electric dipole, the polarizability, and the hyperpolarizabilities:

$$\left.\frac{\mathrm{d}E}{\mathrm{d}\mathbf{F}}\right|_{\mathbf{0}} = -\mathbf{d}_{\mathbf{0}},\ \left.\frac{\mathrm{d}^2E}{\mathrm{d}\mathbf{F}^2}\right|_{\mathbf{0}} = -\boldsymbol{\alpha},\ \left.\frac{\mathrm{d}^3E}{\mathrm{d}\mathbf{F}^3}\right|_{\mathbf{0}} = -\boldsymbol{\chi}^{(3)}, \dots \qquad (3)$$

The derivatives with respect to nuclear coordinates $\mathbf{R}$ include the forces on the nuclei and the force constants, while mixed derivatives can provide information such as Raman intensities [2].

On a classical computer, an energy derivative can be evaluated either numerically or analytically, and we discuss each approach in turn.

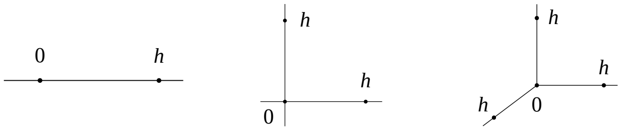

Figure 1: Obtaining a numerical gradient of a function defined on a d-dimensional space classically requires sampling the function $d+1$ times, once at the origin and once at a distance h along each of the axes. Shown above are the sample points for the cases $d=1$ through $d=3$. The quantum gradients algorithm can obtain the same numerical gradient in a single call to the function.

Numerical derivative techniques rely on computing the value of the energy at several discrete points, and then using those values to estimate the true derivative. The simplest technique is finite difference, which for the first derivative in one dimension (and using only two points) is the familiar formula

$$\left.\frac{\mathrm{d}E}{\mathrm{d}\mu}\right|_0 \approx \frac{E(h)-E(0)}{h}. \tag{4}$$

While numerical gradient techniques usually require minimal effort to implement, they are often unreliable because of the ill-posedness of numerical differentiation in general [12]. The problem is essentially that two functions can be uniformly close together while their derivatives are arbitrarily far apart. This usually manifests itself in the fact that small errors in the evaluated functions can lead to large errors in a numerically computed derivative. The problem can be regularized, for example by interpolation or smoothing, but this requires additional assumptions about the numerical data—such as knowledge of an upper bound on the error—which may not be available *a priori*. In particular, methods of *ab initio* electronic structure usually involve long calculations with many potential sources of error, such as rounding, quadrature, or even errors inherent to the approximations being used.

By contrast, analytic derivative techniques are those that compute the derivative by direct evaluation of an analytic expression. They were introducted in quantum chemistry by Pulay [1], and have since almost entirely supplanted numerical procedures. They do not suffer from the numerical instability and they are usually faster as well.

Analytic gradient formulas exist for just about all electronic structure techniques and for mosts kinds of perturbations. To illustrate the argument and establish the correct scaling, we will describe the particularly simple case of derivatives of fully variational wavefunctions. We start by writing the molecular energy as a function $E(\boldsymbol{\mu},\boldsymbol{\lambda}(\boldsymbol{\mu}))$ of the external perturbation $\boldsymbol{\mu}$ and the wavefunction parameters $\boldsymbol{\lambda}(\boldsymbol{\mu})$. These parameters, such as the configuration interaction coefficients or the coupled cluster amplitudes, completely describe the electronic wavefunction. Although $\boldsymbol{\lambda}(\boldsymbol{\mu})$ is a function of $\boldsymbol{\mu}$, for simplicity we will write only $\boldsymbol{\lambda}$. The energy is said to be *fully variational* with respect to $\boldsymbol{\lambda}$ if, given a certain $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$ assumes the value $\boldsymbol{\lambda}^*$ such that the variational condition holds:

$$\left.\frac{\partial E(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}}\right|_* = \mathbf{0}, \tag{5}$$

where $*$ indicates $\boldsymbol{\lambda}=\boldsymbol{\lambda}^*$. In that case we can write $E(\boldsymbol{\mu}) = E\left(\boldsymbol{\mu},\boldsymbol{\lambda}^*\right)$.

For fully variational wavefunctions, the gradient with respect to $\boldsymbol{\mu}$ is given by

$$\begin{aligned}\frac{\mathrm{d}E(\boldsymbol{\mu})}{\mathrm{d}\boldsymbol{\mu}} = \left.\frac{\partial E(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}}\right|_* + \left.\frac{\partial E(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}}\right|_* \frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\mu}} = \\ = \left.\frac{\partial E(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}}\right|_* = \left\langle \boldsymbol{\lambda}^* \left| \frac{\partial H}{\partial \boldsymbol{\mu}} \right| \boldsymbol{\lambda}^* \right\rangle\end{aligned} \tag{6}$$

where we have used the variational condition and the Hellman-Feynman theorem. Since one need not know the first-order wavefunction response $\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\mu}}$, computing the gradient is, to within a constant factor of two or three [6], as hard as computing the energy. That is, once $|\boldsymbol{\lambda}^*\rangle$ is available, the expectation value of the Hamiltonian has approximately the same computational cost as its derivative. However, computing the second derivative (the Hessian) does require the knowledge of the first-order response. In fact, as a direct consequence of Wigner's $2n+1$ rule of perturbation theory, one needs to know the first n responses in order to calculate the $(2n+1)^{\text{th}}$ derivative. Computing the responses often becomes the bottleneck, and it is what leads to a higher asymptotic cost of higher-order derivatives. While the gradient requries about the same resources as the energy, the second and third derivatives require resources that scale as $O(d)$ times the cost of computing the energy (where d is the number of degrees of freedom, i.e., the dimension of $\boldsymbol{\mu}$) [6]. This scaling comes about because $O(d)$ time is required to compute the matrix $\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\mu}}$. Likewise, the scaling of the n^{th} derivative is $O(d^{\lfloor n/2 \rfloor})$, a consequence of the $2n+1$ rule and the $O(d^n)$ cost of computing the n^{th} order response.

The fact that the scaling of derivative techniques, both numerical and analytical, depends on d has meant that these techniques are often restricted to small systems. This is most acutely true of the Hessian, which is often beyond reach, even though the gradient is routinely accessible. We now show that if quantum computers were available, the cost of the higher derivatives would no longer be prohibitive.

THE QUANTUM ALGORITHM

The quantum algorithm for molecular properties is based on Jordan's quantum gradient estimation algo-

	Classical		Quantum
Derivative	Numerical	Analytical	Numerical
$\frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\mu}}$	$O\left(d\right)$	$O\left(1\right)$	$O(1)$
$\frac{\mathrm{d}^2E}{\mathrm{d}\boldsymbol{\mu}^2}$	$O\left(d^2\right)$	$O\left(d\right)$	$O(1)$
$\frac{\mathrm{d}^3E}{\mathrm{d}\boldsymbol{\mu}^3}$	$O\left(d^3\right)$	$O\left(d\right)$	$O(1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\frac{\mathrm{d}^nE}{\mathrm{d}\boldsymbol{\mu}^n}$	$O\left(d^n\right)$	$O\left(d^{\lfloor n/2\rfloor}\right)$	$O(1)$

Table I: Scaling of the time resources required by various techniques of computing molecular properties, in terms of the cost of computing the energy (for example, $O(d)$ scaling means that computing the property requires resources on the order of d evaluations of the molecular energy). E is the total electronic energy, $\boldsymbol{\mu}$ is the external perturbation, and d is the dimension of $\boldsymbol{\mu}$. All the derivatives are evaluated at $\boldsymbol{\mu} = \mathbf{0}$. On classical computers, the numerical scalings correspond to the simplest finite difference scheme. Analytical techniques are the ones that evaluate the derivative directly (the exponent $\lfloor n/2 \rfloor$ comes from Wigner's $2n+1$ rule). On a quantum computer, the computational cost of any derivative is proportional to the cost of computing the molecular energy.

rithm [13, 14]. Jordan's method can numerically compute the gradient of any function F, given a black box that computes the value of F for an arbitrary input. In particular, the algorithm can evaluate the gradient using a a *single* query to F, regardless of the number of dimensions d of the domain of F. By contrast, the simplest classical finite-difference scheme would require $d+1$ queries to F (see Fig. 1). We apply Jordan's algorithm to the computation of molecular properties by specifying a way to compute the energy on a quantum computer as well as by outlining how to obtain higher derivatives. In this section, we describe the algorithm, its application to quantum chemistry, and finally argue that a return to numerical techniques for molecular properties would be justified if quantum computers became feasible.

We assume that the molecular energy is a smooth function $E : \left[-\frac{h}{2}, \frac{h}{2}\right]^d \to [0, \varepsilon)$, where h is chosen sufficiently small, such that E varies sufficiently slowly over the domain. Furthermore, ε is a bound on E. For convenience, we are working in units such that h and ε are unitless and where each dimension has been rescaled as necessary to ensure that the bounds are the same along all of the axes. Jordan's algorithm starts in an equal superposition on d registers of n qubits each (nd qubits total):

$$\frac{1}{\sqrt{N^d}} \sum_{k_1=0}^{N-1} \cdots \sum_{k_d=0}^{N-1} |k_1\rangle \cdots |k_d\rangle = \frac{1}{\sqrt{N^d}} \sum_{\mathbf{k}} |\mathbf{k}\rangle\,, \qquad (7)$$

where $N = 2^n$, the states $|k_i\rangle$ are integers on n qubits represented in binary notation, and $|\mathbf{k}\rangle$ is a d-dimensional vector of all the $|k\rangle$'s. Assuming that we have a black box for E (discussed below), one can use phase kickback to prepare, with a single call to the black box,

$$\frac{1}{\sqrt{N^d}} \sum_{\mathbf{k}} \exp\left[2\pi i \frac{N}{h\varepsilon} E\left(\frac{h}{N}\left(\mathbf{k} - \mathbf{N}/2\right)\right)\right] |\mathbf{k}\rangle \approx$$
$$\approx \frac{1}{\sqrt{N^d}} \sum_{\mathbf{k}} \exp\left[2\pi i \frac{N}{h\varepsilon}\left(E(\mathbf{0}) + \frac{h}{N}\left(\mathbf{k} - \mathbf{N}/2\right) \cdot \frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\mu}}\right)\right] |\mathbf{k}\rangle\,, \qquad (8)$$

where $\mathbf{N}$ is the vector $(N, N, \ldots, N)$, and the approximation is valid for sufficiently small h (the error caused by quadratic terms and higher is discussed in Jordan's paper[13], but is in any case only polynomial). The final state is separable and equals

$$\frac{\exp\left[2\pi i \frac{N}{h\varepsilon} E(\mathbf{0})\right]}{\sqrt{N^d}} \sum_{k_1=0}^{N-1} \exp\left[\frac{2\pi i}{\varepsilon} k_1 \frac{\partial E}{\partial \mu_1}\right] |k_1\rangle \cdots$$
$$\cdots \sum_{k_d=0}^{N-1} \exp\left[\frac{2\pi i}{\varepsilon} k_d \frac{\partial E}{\partial \mu_d}\right] |k_d\rangle\,, \qquad (9)$$

whereupon we can apply the inverse quantum Fourier transform to each of the d registers and measure in the computational basis to obtain the gradient (with a global phase and scaled by a factor of N/ε),

$$\exp\left[2\pi i \frac{N}{h\varepsilon} E(\mathbf{0})\right] \left|\frac{N}{\varepsilon}\frac{\partial E}{\partial \mu_1}\right\rangle \cdots \left|\frac{N}{\varepsilon}\frac{\partial E}{\partial \mu_d}\right\rangle =$$
$$= \exp\left[2\pi i \frac{N}{h\varepsilon} E(\mathbf{0})\right] \left|\frac{N}{\varepsilon}\frac{\partial E}{\partial \boldsymbol{\mu}}\right\rangle. \qquad (10)$$

The number of bits of precision equals n, which can be made as large as necessary. It should be reiterated that a single call to E was made, as opposed to the $d+1$ that would be needed in the case of numerical differentiation by finite difference.

Ignoring various scaling factors, the gradient estimation algorithm produces the transformation

$$|\mathbf{0}\rangle \longrightarrow e^{iE(\mathbf{0})} \left| \left.\frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\mu}}\right|_{\mathbf{0}} \right\rangle. \qquad (11)$$

We can compute the Hessian (and higher derivatives) by iterating this algorithm. If, instead of making a call to $E(\boldsymbol{\mu})$, the algorithm sought $E(\boldsymbol{\mu} - \mathbf{a})$ from the oracle, we would perform, at the cost of a single additional subtraction,

$$|\mathbf{0}\rangle \longrightarrow e^{iE(\mathbf{a})} \left| \left.\frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\mu}}\right|_{\mathbf{a}} \right\rangle. \qquad (12)$$

Another evaluation of E would suffice to remove (or "uncompute") the global phase. This means that we have designed an oracle that, given $\mathbf{a}$, yields $\left| \left.\frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\mu}}\right|_{\mathbf{a}} \right\rangle$ and uses only two calls to the "original" oracle for E. Using the gradient algorithm with this new oracle, we obtain the state (again ignoring scaling factors)

$$e^{i\,\frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\mu}}\big|_{\mathbf{0}}}\left|\left.\frac{\mathrm{d}^2E}{\mathrm{d}\boldsymbol{\mu}^2}\right|_{\mathbf{0}}\right\rangle, \qquad (13)$$

which is a two-dimensional array of d^2 quantum registers containing all the elements of the Hessian of E. Computing higher derivatives would require additional factors of two in the number of required oracle calls, caused by the need to uncompute a global phase at each step (this problem is a common feature when it comes to recursing quantum algorithms—see e.g. [15]). Hence, evaluating the n^{th} derivative requires 2^{n-1} queries to E, which, although exponential in n, is much better than d^n+1, which is the minimum number of function queries required to compute the derivative by classical finite difference. We stress that the number of calls to E is independent of d, and thus of the size of the system, for the derivative of any order.

One could object that the quantum gradient algorithm is a numerical approach and therefore suffers from the same numerical instability problems as the classical numerical gradient techniques. This claim is true, and it implies that the quantum gradient algorithm cannot be used indiscriminately for problems that feature uncontrolled sources of error. This would include, in particular, many quantum chemistry techniques, meaning that this algorithm could not be coupled to any classical electronic structure method. However, if the technique for computing the energy is numerically exact, that is to say, if the error in the energy can be controllably reduced below any level, the magnitude of the numerical error in the calculated derivative can likewise be made arbitrarily small. After all, numerical techniques could safely be used on classical computers as well, if only exact electronic structure methods, such as full configuration interaction (FCI), were feasible for molecules with more than a few atoms.

Fortunately, quantum computers would make it possible to evaluate the molecular energy in polynomial time, meaning that numerical instability should not be a problem. We turn to this topic next.

THE ENERGY BLACK BOX

The application of Jordan's gradient algorithm to chemical problems requires that there be a black box that can compute the value of the ground-state molecular energy at any value of the perturbation $\boldsymbol{\mu}$ in the neighborhood of $\boldsymbol{\mu}=\mathbf{0}$. Furthermore, to avoid numerical artifacts, this black box should be numerically exact, allowing the error in the energy to be controllably reduced at will.

The problem of exact classical electronic structure methods is that they generally have a computational cost that scales exponentially with the size of the system. Although these classical algorithms could also be used as subroutines in the quantum gradient algorithm, there are quantum electronic structure algorithms that could avoid the exponential scaling altogether.

In particular, we have recently described a quantum full CI algorithm [7] for computing the molecular ground state energy in $O\left(M^5\right)$ time [16], where M is the number of basis functions. This algorithm could be easily made into a subroutine that would function as the black box for the energy. Several modifications would have to be made, including an improvement to the algorithm that would allow it to compute all the overlap integrals on the quantum computer, thus rendering it possible to introduce the perturbation $\boldsymbol{\mu}$ into the calculation. Nevertheless, a quantum computer running the quantum FCI algorithm could be used to obtain any molecular property of a system with basis size M in $O\left(M^5\right)$ time, a dramatic improvement over the possibilities of classical computers.

A more recent development is the real-space chemical dynamics simulation algorithm [9, 17], based on the earlier work of Zalka and Wiesner [18, 19]. It is known that simulating the exact dynamics of a system of P particles interacting under a pair-wise interaction requires at most $O(P^2)$ time and $O(P)$ space, in contrast to the classical exponential cost. If an eigenstate of the system Hamiltonian were prepared as the initial state [20], the dynamics would only apply a phase to the wavefunction. This phase could be read out by the phase estimation algorithm [21, 22], forming the required energy black box. Although this algorithm requires higher overhead than the equivalent quantum FCI calculation, it benefits from a superior asymptotic scaling as well as from the fact that only minimal modifications would need to made to insert the perturbation $\boldsymbol{\mu}$ into the calculation. For example, simulations with different nuclear coordinates proceed in exactly the same way, while an electromagnetic field requires only a small modification of the simulated Hamiltonian [18].

NEWTON'S METHOD AND GEOMETRY OPTIMIZATION

Perhaps the single most common use of molecular derivatives is molecular geometry optimization. We can therefore use it to illustrate some of the advantages of a quantum algorithm over a classical one, including a quantum version of Newton's method, which offers an aditional quadratic speedup over its classical counterpart.

A simple way for finding the locally optimal geometry is to perform the standard Newton iterations,

$$\mathbf{R}_{n+1}=\mathbf{R}_n-\left(\left.\frac{\mathrm{d}E}{\mathrm{d}\mathbf{R}}\right|_{\mathbf{R}_n}\right)\cdot\left(\left.\frac{\mathrm{d}^2E}{\mathrm{d}\mathbf{R}^2}\right|_{\mathbf{R}_n}\right)^{-1}, \qquad (14)$$

until convergence is reached. Here, $\mathbf{R}_n$ are the nuclear coordinates at the n^{th} iteration, and $\frac{\mathrm{d}E}{\mathrm{d}\mathbf{R}}\big|_{\mathbf{R}_n}$ and $\frac{\mathrm{d}^2E}{\mathrm{d}\mathbf{R}^2}\Big|_{\mathbf{R}_n}$ are, respectively, the gradient and Hessian of E with respect to nuclear displacement (the "molecular gradient" and the "molecular Hessian"). If a quantum computer were used to compute the derivatives, one would require exactly 3 calls to a black box for E per iteration: one for the gradient and two for the Hessian. A classical approach, on the other hand, would be much slower, requiring at least d^2+1 function calls for finite difference, and approximately $O(d)$ effort in the analytical case [29]. For large molecules with large d, this savings could prove significant, even if each energy evaluation takes much longer on a quantum machine than on a classical computer.

There are many classical tricks available for speeding up the convergence of Newton's method if the initial guess is not close to a local minimum, in which case the usual Newton step might be inappropriately large. Techniques such as trust regions and level shifts [23] are still available to quantum computers, or they can be implemented as classical post-processing.

In addition, we remark that Newton's method is the first in the class of Householder methods, which offer a rate of convergence of $\ell+1$, provided that derivatives up to order $\ell+1$ exist and can be calculated. A quantum computer could be used to accelerate Householder methods of any degree, requiring $\sum_{m=1}^{\ell+1} 2^{m-1} = 2^{\ell+1}-1$ calls to the black box for order-ℓ Householder optimization method. Although exponential in ℓ, this expression is independent of system dimension d.

Of course, Newton's method is only useful for local minimization, and we are often interested in global optimization. Here, we can use a quantum version of the multistart technique, called the quantum basin hopper [24–27]. A number of points is selected at random, and each is followed, using a local search, to its local basin (if a quantum version of Newton's method is used for the local search, such as the one we propose above, we can get the usual quadratic convergence). Then, the minima of all the basins are compared and the least one chosen as the global minimum. Quantum computers could add a quadratic speed-up to such a multistart technique, since the resulting local minima form an unstructured database that can be searched using Grover's algorithm [10]. As Dürr and Høyer pointed out [28], a Grover search can find the minimum of an unstructured database with $O(\sqrt{K}\log K)$ calls to the database (where K is the number of database entries, i.e. multistart points), as opposed to the classically required $O(K\log K)$ queries.

We acknowledge support from the Army Research Office under contract W911NF-07-0304. IK thanks the Joyce and Zlatko Baloković Scholarship.

[*] Electronic address: aspuru@chemistry.harvard.edu

[1] P. Pulay, Molecular Physics **17**, 197 (1969).

[2] P. Pulay, Advances in Chemical Physics **69**, 241 (1987).

[3] T. Helgaker and P. Jørgensen, Advances in Quantum Chemistry **19**, 183 (1988).

[4] Y. Yamaguchi, J. D. Goddard, Y. Osamura, and H. Schaefer, *A New Dimension to Quantum Chemistry: Analytic Derivative Methods in Ab Initio Molecular Electronic Structure Theory* (Oxford University Press, 1994).

[5] R. Shepard, in *Modern Electronic Structure Theory*, edited by D. Yarkony (World Scientific, 1995), p. 345.

[6] P. Pulay, in *Modern Electronic Structure Theory*, edited by D. Yarkony (World Scientific, 1995), p. 1191.

[7] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Science **309**, 1704 (2005).

[8] H. Wang, S. Kais, A. Aspuru-Guzik, and M. R. Hoffmann, Phys. Chem. Chem. Phys. **10**, 5388 (2008).

[9] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, Proc. Natl. Acad. Sci. **105**, 18681 (2008).

[10] L. Grover, Proceedings of the 28th Annual ACM Symposium on the Theory of Computing pp. 212–219 (1996).

[11] T. Helgaker, in *European Summerschool in Quantum Chemistry Lecture Notes*, edited by P.-O. Widmark (Lund University, 2007), p. 383.

[12] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems* (Vh Winston, 1977).

[13] S. P. Jordan, Phys. Rev. Lett. **95**, 050501 (2005).

[14] D. Bulger, quant-ph/0507109 (2005).

[15] S. Aaronson, Quant. Inform. Comput. **3**, 165 (2003).

[16] B. P. Lanyon, J. D. Whitfield, G. G. Gillet, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, et al., arXiv:0905.0887 (2009).

[17] D. A. Lidar and H. Wang, Phys. Rev. E **59**, 2429 (1999).

[18] C. Zalka, Proc. Roy. Soc. A **454**, 313 (1998).

[19] S. Wiesner, quant-ph/9603028 (1996).

[20] N. J. Ward, I. Kassal, and A. Aspuru-Guzik, J. Chem. Phys. **130**, 194105 (2009).

[21] A. Y. Kitaev, quant-ph/9511026 (1995).

[22] D. S. Abrams and S. Lloyd, Phys. Rev. Lett. **83**, 5162 (1999).

[23] R. Fletcher, *Practical Methods of Optimization* (Wiley, 2000), 2nd ed.

[24] D. Bulger, quant-ph/0507193 (2005).

[25] D. Bulger, J. Optim. Theory Appl. **133**, 289 (2007).

[26] W. P. Baritompa, D. Bulger, and G. Wood, SIAM J. Optim. **15**, 1170 (2005).

[27] J. Zhu, Z. Huang, and S. Kais, arXiv:0906.2375 (2009).

[28] C. Dürr and P. Høyer, quant-ph/9607014 (1996).

[29] Quasi-Newton methods (such as the Broyden–Fletcher–Goldfarb–Shanno method), or even simpler methods such as simple gradient descent, can remove the need to compute the molecular Hessian at each step, or at all. While such schemes are useful and widely applied, we do not discuss them here because they are typically slower and less acurate than Newton's method. While they offer a classical computational advantage, on a quantum computer that advantage would be erased by the ability to rapidly compute the exact Hessian at each step.