Adiabatic Quantum Computation and Simulation

December 8, 2005

Vlad Goldenberg, Maciek Sakrejda

1 Introduction

Adiabatic quantum computation came into the spotlight of quantum information processing researchers several years ago. The idea suggested an alternative approach to quantum computing which can be shown to be equivalent to the earlier quantum circuit analysis, but introduced possible convenience advantages over implementing the circuit model. Recently the adiabatic approach to quantum computing has had dramatic up and downturns as a viable way to approach quantum computing, along with related controversy regarding the consistency of the adiabatic theorem.

1.1 Quantum Adiabatic Theorem

The adiabatic theorem of quantum mechanics states that a system in an nth eigenstate evolving under a Hamiltonian will remain in the nth eigenstate throughout the evolution provided the neighborhood energy gaps are small enough and the evolution is carried out slowly enough'.

That is, let $\hat{H}_0$ be the Hamiltonian of our system in the initial state. Associated with this initial initial Hamiltonian are instantaneous eigenstates, denoted by $|n_0\rangle$. In general, let $\hat{H(t)}$ be an evolving Hamiltonian at time t, and it's eigenstates call $|n(t)\rangle$ such that:

$$\hat{H}(t) = |n(t)\rangle$$

First, allow T to be the final time for the Hamiltonian evolution, and make the substitution $s = \frac{t}{T}$. Then it is possible to design a linearly varying Hamiltonian such that

$$\hat{H}(s) = \hat{H}_0(1-s) + \hat{H}_1(s)$$

so that at time$t = 0$ the Hamiltonian is in the initial state and at time $t = T$ the Hamiltonian is in the final state of the system. (Ali2004) suggests that a

stringent critereon for the time evolution of the Hamiltonian are the following:

$$\delta_{min} = min_{0 \leq s \leq 1}[E_j(s) - E_k(s)] \tag{1}$$

$$\Delta_{max} = max \left\| \frac{d}{ds} \hat{H}(s) \right\| \tag{2}$$

$$T = \frac{\Delta_{max}}{\epsilon \delta_{min}^2} \tag{3}$$

These formulas put a minimum value on how long (T) must be for the final state to be ϵ close to the ground state.

It is possible to design the initial and final Hamiltonians such that the ground eigenstates are the input and desired output of the computation. (Aharanov2005) points out that if the final Hamiltonian encodes the result of the computation as it's ground state, but the ground state determines what Hamiltonian is needed, then the output state must be known before the computation is accomplished, which beats the point of using the adiabatic model. They point out, however. that nothing need be known about the final output state because it can be shown that as long as the ground state of the final Hamiltonian has a non-negligible inner product with the result of the computation, it is still possible to perform a quantum computation adiabatically.

1.2 Adiabatic vs Circuit Model

(Ali2004) suggests that it is not necessary to be in the ground state to perform a calculation. It is argued that it is sufficient that by properly designing the evolution of the Hamiltonian to have the effect of applying an instantaneous gate, a qubit gate can be performed adiabatically, with the result of the computation *not* necessarily being in the ground state of the final Hamiltonian. (Aharanov2005) showed that any quantum circuit model can be implemented with an adiabatic model with polynomial time, and conversely, any adiabatic algorithm can be simulated with a quantum circuit model, and therefore concluded that the two approaches are equivalent.

2 Implementation

Various papers have suggested seperate methods for implementing adaibatic quantum computation, including (Aharanov2005) ground state implementation and the (Ali2004) two-state system implementation. The specific implementation of (Ali2004) shall be discussed.

2.1 2-state Single Qubit Gate

Let us say that we would like to implement the single qubit Hadamard gate using adiabatic computation. We know that the transformation is represented

by

$$H = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Therefore the action of H on the computational basis is:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Now we can let our initial and final Hamiltonians be the projectors of the basis states and their corresponding outputs with corresponding eigenvalues. That is,

$$\hat{H}_0 = -E|0\rangle\langle 0| + E|1\rangle\langle 1|$$
$$\hat{H}_1 = -\frac{E}{2}(|0\rangle + |1\rangle)(\langle 0| + \langle 1|) + \frac{E}{2}(|0\rangle - |1\rangle)(\langle 0| - \langle 1|)$$

If we follow the previous prescription for linear Hamiltonian evolution,

$$\hat{H}(s) = \hat{H}_0(1-s) + \hat{H}_1(s)$$

We see that since, according to the adiabatic theorem, the ground eigenstate should remain in the instantaneous ground state, and the first excited state should remain in the instantaneous first excited state, the final evolution will be equivalent to the circuit model of the single-qubit Hadamard gate.

3 Satisfiability

Another interesting problem to which we can apply adiabatic quantum computation is the satisfiability question from complexity theory, specifically the 3-SAT problem. Unfortunately, the adiabatic quantum approach has been shown to take exponential time in the worst case, as do all known classical algorithms, but this is nevertheless an interesting problem to consider in order to illustrate the workings of the adiabatic approach.

The 3-SAT problem consists of n bits in M clauses of 3 bits each, where we can reuse bits across clauses:

$$C_1 \wedge C_2 \wedge \ldots \wedge C_M$$

Each clause has a satisfying assignment that depends on the value of its three bits, and the 3-SAT problem as a whole has a satisfying assignment if and only if all of its clauses are satisfied. As all adiabatic quantum problems, we approach this one by specifying a time-dependent Hamiltonian and an initial state. The Hamiltonian takes the form of a linear combination of Hamiltonians, each acting on a separate 3-qubit clause (and only on the qubits in that clause):

$$H(t) = H_{C1}(t) + H_{C2}(t) + \ldots + H_{CM}(t)$$

As always, we define t as slowly varying between zero and T, and the initial state as the ground eigenstate of $H(0)$. For each clause, the ground state of its corresponding Hamiltonian, at time T, encodes the satisfying assignments of the clause, and therefore the total Hamiltonian $H(T)$ encodes the solution to the satisfiability problem in its ground state.

We can define an energy function h_C for a clause such that the energy is 0 if the assignment satisfies the clause and 1 otherwise. The total energy h is the sum of these terms. Thus, h is zero if all clauses are satisfied; otherwise, it is greater than zero. We can then use this energy to define the final Hamiltonian, H_f, as a sum of the individual H_{fC} terms that act on the qubits in each clause, and where each H_{fC} acts by multiplying the state by its corresponding energy (as given by h_C). We can see that the final Hamiltonian is nonnegative and that it is equal to zero only when operating on a superposition state of qubits that satisfy all constituent clauses.

We then define the initial Hamiltonian to be the sum of a set of clause-specific Hamiltonians, where each clause Hamiltonian is acting on the clause's three qubits, and each individual qubit Hamiltonian is

$$H(i) = \frac{1}{2}(1 - \sigma_x),$$

with σ_x being the Pauli x operator acting on i. We then choose our initial state to be the eigenstate to this Hamiltonian's lowest eigenvalue, which is the superposition state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

We can now construct the total Hamiltonian as a linear combination of H_{inital} and H_{final}, as before. We then consider the time T it will take for this evolution to take place. We consider a solution feasible if the time T required for the adiabatic evolution is at most polynomial in n, the number of qubits we are working with. Looking at the requirements for adiabatic evolution, we see that

$$T \gg \frac{\epsilon}{g_{min}^2}$$

where ϵ is the matrix element between the lowest two eigenstates. Given our formula for the total Hamiltonian, we know that this is just $H_{final} - H_{initial}$, which means that epsilon can be no larger than the maximum eigenvalue of this operator. Since the spectrum of H_{final} is contained in {1, 2, ..., M} (where M is the total number of clauses), and the spectrum of H_{inital} in {1, 2, ..., d} (where d is the sum total of all the clauses in which each bit appears). In 3-SAT, d is clearly bounded by 3M, which occurs when each clause consists of three unique bits (although, of course, reusing the same bits is allowed across clauses). The value of g_{min} itself, however, is trickier and cannot be deduced so easily. It can be worked out for specific problem instances, but in general, it can be exponentially small, leading to an exponential timing schedule T. Unfortunately, the problems where g_{min} is not exponentially small are generally those that are easy to solve classically. It is unclear whether there are computationally difficult satisfiability problems with a polynomial timing schedule. Since Satisfiability is an NP-Complete problem, any NP-hard problem is polynomially equivalent

to it, and an efficient quantum adiabatic algorithm for this problem would be a great discovery in complexity theory. However, this also means that such an algorithm for the general-case satisfiability (or even specifically 3-SAT) problem is unlikely.

4 Simulation

Simulation of quantum adiabatic systems is interesting not only for what it can tell us about the behavior of the actual algorithms, but also for the physical implementation issues it brings up. Classical simulation requires a reduction of the problem to strictly formal terms, and exposes many issues which can easily be glossed over in a less rigorous analysis.

One of the most obvious issues around simulation is the lack of quantum properties in a classical computer simulating a quantum algorithm. This is not really a problem, however, since relevant quantum properties, such as superposition and entanglement, can be simulated (though of course at computational cost prohibitive for anything but a simulation). This does mean that classical simulations are limited to quantum systems with relatively few qubits: the underlying classical computations are mostly matrix operations, which are rather computationally expensive.

An important issue this brings up is the need for local Hamiltonians. A local (or k-local) Hamiltonian is one which can be described as the sum of smaller Hamiltonians, each of which acts on at most k qubits. We need a k-local Hamiltonian because, in a simulation, a Hamiltonian acting on every qubit of an n-qubit state takes $2^n \times 2^n$ space and, more importantly, $O(n^3)$ time per step. These are very real constraints that translate to a quantum implementationit would be physically very difficult to arrange a quantum circuit in which every qubit acts on every other qubit.

Another issue is discretization. Classical computers are, by design, discrete-based on two states, zero and one. Any seemingly continuous computation is just an approximation, and rounding errors must be carefully compensated for in order to avoid drifting from, for example, the true path of a function by accumulating errors. That is, if we naively plot y = 1/3 x by just adding 1/3 to y every 1 unit in x, and the underlying representation of 1/3 is not exactly accurate, after a while, our function will begin to noticeably sag. This must be carefully accounted for in adiabatic evolution, where clearly following the ground energy eigenvalue is key to successfully following the algorithm.

A related issue is the resolution in a simulation: if our steps are too large, we may miss the most constraining part of the spectral gap and not even realize it. Suppose we are following the path of the ground eigenvalue from 0 to T. Suppose that at some point, it has a local maximum that crosses paths with a local minimum for the first excited eigenvalue. Now suppose that this local maximum is surrounded by two inflection points. If these are all very close together, and if our steps are large enough (that is, too large), we may end up at the inflection point to the left of the maximum, evaluate the slope of the eigenvalue, take a

step, and end up at the other inflection point, having missed the maximum (and hence the intersection with the path of the first excited eigenvalue) entirely. The algorithm would produce garbage output, or worse, output that looked reasonable, but in fact did not correspond to reality. More interestingly, even if we do not hit a crossing, and we merely miss the most constraining part of the spectral gap, we still will have moved too fast to get through the gap safely (that is, we will have violated the premises of the adiabatic theorem regarding the timing schedule), and a real physical implementation would have jumped eigenvalues in such a situation. Since the entire point of a simulation is to represent what would happen if a system were actually implemented, this is clearly undesirable.

This brings up more spectral gap issues. If we naively follow a fixed step size throughout the entire algorithm, we are over-constraining the step size, and our computation may take longer, often by an additional order of complexity. The best demonstration of this is the quantum adiabatic solution to Grover's problem. The nave approach, finding the minimum gap over the entire progress of the algorithm and using this as a basis for a constant step size, leads to an O(N) performance time, similar to classical algorithms. However, instantaneously adjusting the rate of progress according to the minimum gap at any given moment lets us recover the square root speedup achieved by other, non-adiabatic, quantum algorithms. This is somewhat difficult in a classical simulation as it is, strictly speaking, impossible, to take infinitesimal steps in a discrete environment. However, varying the step size dynamically by constantly re-examining the spectral gap and re-adjusting the step size is usually a sufficient approximation.

Yet another issue arises: sometimes, our eigenvalue levels do cross, or get exponentially close, when using just the nave Hamiltonian, a linear combination of Hinitial and Hfinal. It is sometimes possible to recover a Hamiltonian where eigenvalue paths do not cross by adding a non-linear perturbation to the linear combination Hamiltonian. That is, we pick an A such that

$$\hat{H}(s) = (1s)\hat{H}_i(s) + s\hat{H}_f(s) + As(1s)\hat{H}_{aux}(s)$$

has eigenvalue levels which do not cross as s goes from 0 to T, where Haux is an auxiliary Hamiltonian that does not affect the initial or final state (as can be seen from the s(1 s) coefficient), but is only present to regulate the eigenvalue levels. This term is specifically constructed to avoid crossing levels and has no bearing on the meaning of the initial or final state.

These are just some of the issues that come up when making a deeper examination of a quantum adiabatic system in preparation for simulation.

5 Conclusions

Quantum adiabatic computation provides an interesting new paradigm in algorithmic problem-solving. It has been shown that it can be simulated, with a polynomial slowdown, by a more traditional circuit-based quantum system

of unitary gates. This means that it obviously cannot provide a speedup over such a system. However, adiabatic quantum computation has been shown to be competitive with a circuit-based model on a wide range of algorithms, and the method itself encourages new ways of thinking about computation, just as quantum algorithms offer a new methodology vis-a-vis classical algorithms. Moreover, with quantum computing hardware still in gestation, it may turn out to be the case that some things are significantly easier to implement as adiabatic algorithms. Overall, while it certainly is not a silver bullet, this is an interesting field in quantum computing that could offer interesting developments.

6 References

Farhi, Edward et. al. Quantum Computation by Adiabatic Evolution, quant-ph/0001106 Jan 2000

Sarandy, M.S., Wu, L.A., and Lidar, D.A., Consistency of the Adiabatic Theorem, Quant. Inf. Proc. Vol. 3, No. 6, Dec. 2004

Roland, Jeremie and Cerf, Nicolas J., Quantum Search by Local Adiabatic Evolution quant-ph/0107015 Jul 2001

Andrecut, M. and Ali, M.K., Adiabatic Quantum Gates, Int. Jour. Th. Phys., Vol. 43, No. 4, April 2004

Aharonov, Dorit, et. al., Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation, quant=ph/0405098 Mar. 2005

van Dam, Wim, Mosca, Michele, Vazirani, Umesh, How Powerful is Adiabatic Quantum Computation?, quant-ph/0206003 Jun 2002