Characterizing Computational Techniques for Experimentally Feasible Non-Local Evolution of Two-Qubit Systems

Shankar Balasubramanian Didi Chang-Park Kyle Herndon Zane Rossi

TJHSST Modern Physics and Optics Lab

Abstract: This paper investigates computational methods and presents a computational framework for simulating non-local evolution of two-qubit systems. This evolution may then easily be combined with finite local operations to give control of near-arbitrary precision. Current research has focused on developing the theoretical background which we hope to computationally implement. We will characterize how useful operators (particularly the *CNOT* gate) are generated in this process, and they will be approximated and performed computationally in optimal time utilizing a adaptive-algorithm approach. We will also analyze the non-local piece of the system Hamiltonian, including determining experimental means of generating it. Particularly, we will be evaluating its action on the Poincaré-Bloch-Sphere.

Mutation Protocol

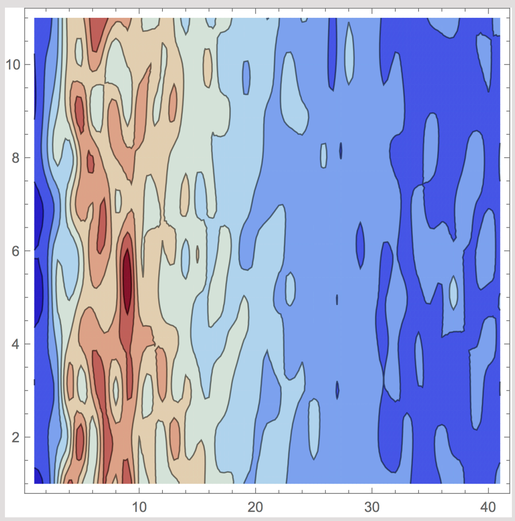


Figure 1: This plot shows the fidelity of the best organism in a ten-organism set after 100 generations with respect to mutation chance and mutation magnitude. The vertical axis, when the value is multiplied by 0.1, gives the chance in any generation for one of the angles defining an organism to be mutated. The horizontal axis, when multiplied by 0.025, gives the mutation magnitude (this ranges between 0 and 1 radian(s)). The plot shows clear correlation between mutation magnitude and organism success.

Mutations are designated on the basis of chance and magnitude. We set a particular threshold value, which we are free to change. This threshold value (α) determines whether or not a particular organism's parameters will be mutated. The parameters, as discussed above, are the β' rotation coefficients which are specified in the exponential form of the local operations. The other changeable parameter, μ, specifies the degree to which the angles will be mutated. These mutations encompass an envelope of random values between $-\mu$ and μ (a uniform distribution) which will be added to the current β' values to update them. Figure 1 depicts the efficacy of the simulation (particularly, the fidelity), for different values of α and μ. Lighter colors represent simulations resulting in higher fidelities. The values of α and μ do not change across the entire simulation; however, the efficiency of our program can be increased by modifying these values per each generation

The Space as a Whole

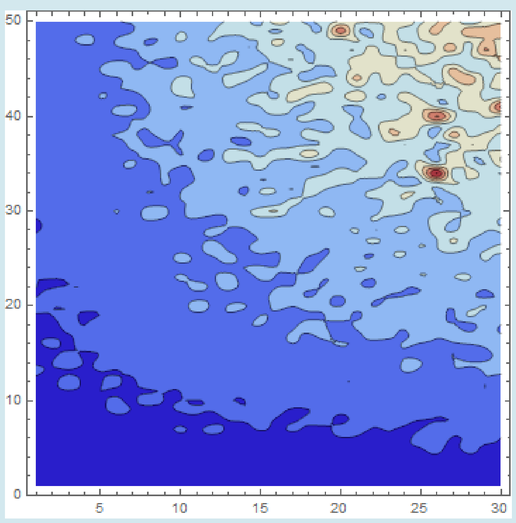
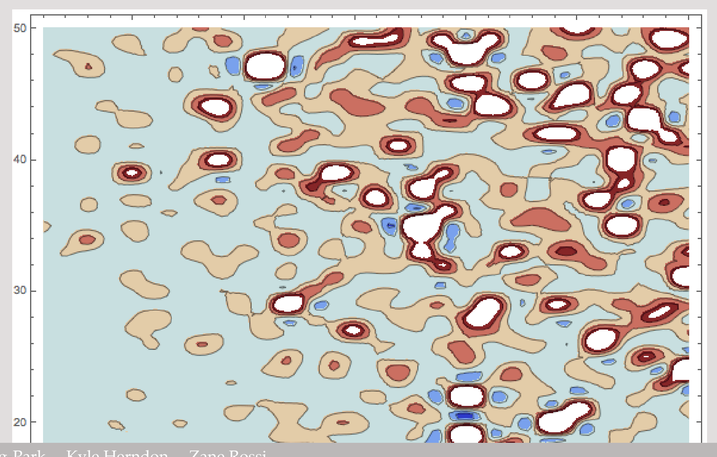


Figure 2: 50 organisms and 30 generations.

XY Problem



Classifying Fidelity and Breeding Operators

We measure the fidelity of a matrix in a very conventional manner

$$\mathcal{F} = \frac{1}{|A - G|} \tag{1}$$

where A is a current matrix, and G is a goal matrix that we hope to reach. The magnitude of a matrix in this case is defined to be the square root of the sum of the squares of its elements. The greater the fidelity, the closer the generated matrix is to the goal matrix.
When we breed two matrices together, we consider their individual fidelities $\mathcal{F}_1$ and $\mathcal{F}_2$. We create the new parameters of a bred matrix based on the fidelities and angular parameters of the two parent matrices. Mathematically, this is given by a weighted average,

$$\beta^i_{breed} = \frac{\mathcal{F}_1\beta^i_1 + \mathcal{F}_2\beta^i_2}{\mathcal{F}_1 + \mathcal{F}_2} \tag{2}$$

where i refers to all the relevant angular parameters. This will, given a sufficiently smooth space, allow the child organisms to retain properties of the parents, and fare better than if their mutations had been random, and each generation was composed of clones of the previous generation's most successful organisms. Note that in Figure 1, the organisms were not subject to breeding. Despite this, we see remarkable success.

Variation in Population

A series of methods is used to determine fidelity in a populations of a set size, generation number, and gene length. We can then produce a series of partially-evaluated functions that can be mapped over a list of integers. When the results of this mapping are plotted, we gain knowledge of a slice of the search space. Knowledge of these slices, analogous to partial derivatives, can help determine which variables produce the best solutions with the smallest computational cost.

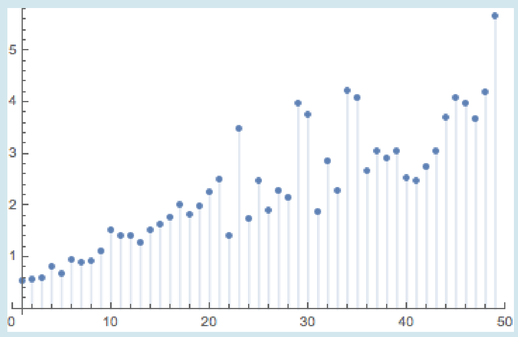


Figure 4: A plot of the average of 20 runs in which the population is along the horizontal axis, and fidelity ($\mathcal{F}$), is along the vertical axis. During these runs, the organism population was brought through 20 generations, with each organism having two genes. The mutation chance (α) and mutation magnitude (μ) are drawn from the maximum of Figure 1 (0.5 and 0.225 respectively).

Here we see a plot of constant generation number and gene length, where population size is being altered. Over a large average, the upward trend is clearly visible, as would make sense, and closely follows an exponential trend. We can conjecture that, given a saturation of the space with organism, we may eventually approach a point of diminishing returns, but that is not apparent in Figure 4. These factors indicate a volatile variable, where we are sacrificing memory for some time and search space-covering.

Variation in Generation Number

The same process is used as described above, save now the slice through the search space is made in a way orthogonal to the original slice. On average, due to the structure of the space, the behavior along one slice is related to that at adjacent slices. We see a similar but not identical behavior.

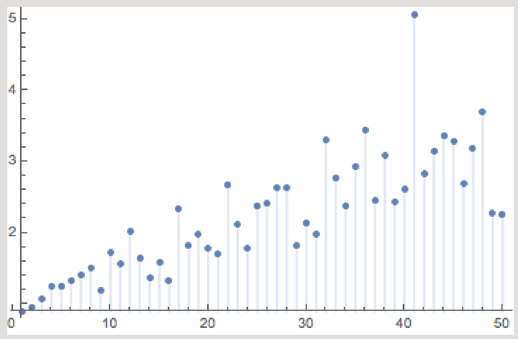


Figure 5: A plot of the average of 20 runs in which the generation number is along the horizontal axis, and fidelity is along the vertical. During this run, the number of generations is taken from 1 to 50, while the gene number remains at 2, and the

Background

The ultimate goal of Control Theory is the ability to arbitrarily manipulate the state of an ensemble of qubits. In the context of quantum computation, this problem is reduced further: basic quantum gates allow for modular systems [3]. However, the generation of these gates is non-trivial, and the arbitrary Hamiltonian is, in general, not readily constructible. Often one will wish to achieve a specific unitary evolution operator or gate U corresponding to the solitary time evolution of a Hamiltonian H. When H may not be engineered easily, the evolution can be achieved in other ways [3] , often involving the use of local operations (LO), repeated measurements on the system [2], along with standard Hamiltonian-driven evolution drawing from some collection (possibly a one element collection) $H_1, H_2, \cdots, H_n$, all of which *can* be lucratively generated.
In the simplest case, a two particle system, the Hamiltonians involved may contain non-local terms, usually corresponding to particle-particle interactions. This paper will propose a framework for taking the general non-local Hamiltonian [2]

$$H = A \otimes \mathbb{1} + \mathbb{1} \otimes B + \sum_{ij} M_{ij}\, \sigma_i \otimes \sigma_j \tag{3}$$

and interspersing its non-local part's evolution with LO to achieve a desired U. It is only the non-local part of the Hamiltonian which must be studied, and through decomposition of this piece into its canonical form, and some insight into Lie Theory, we are able to formulate a computational approach [4]. Devising a scheme to generate a U from a collection of available operators and Hamiltonians has direct applicablity to a variety of problems, but cannot be done purely analytically on a massive scale, thus making a computational method necessary.

Adaptive Algorithm Theory

Adaptive algorithms attempt to mimic the process of evolution by the mechanisms of mutation and natural selection. A set of mathematical objects dubbed *organisms*, each containing some set of data, are subjected to tests against some metric depending on the organism's features [6]. This is considered one *generation*, and the best organisms of each generation are bred together (by some deterministic algorithm) and mutated to compete in the next generation.

Θ : Arguments

H : Constant

N : Constant

$\{A_i\}$ $\{B_i\}$

$\{t_i\}$

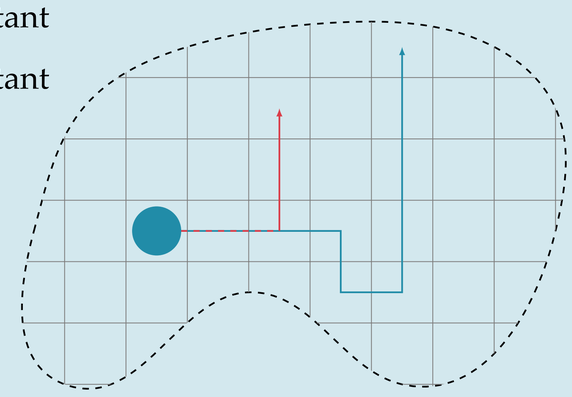


Figure 6: This is a simplification; the organism being tested is represented by the circle in blue, while the search-space (which has a defined metric represented by the grid) surround it. The paths of different color shown represent the non-deterministic paths possible as the organism mutates and is bred with other successful organisms. The actual space being searched is not two- but many-dimensional.

Group Theory Exploitation

The entirety of this simulation scheme is done with the pauli basis, and this allows their wonderful properties to be exploited. Given further research, the algorithm would implement group theory concepts in its heuristic, but currently, this idea is only used in data compression, and the breeding/mutation of matrices.
The local operations described in the preceding section can be realized as unitary matrices, and described by three angles. Because of this, we can imagine all possible unitary operators as lying on the surface of the 3-Sphere[1].

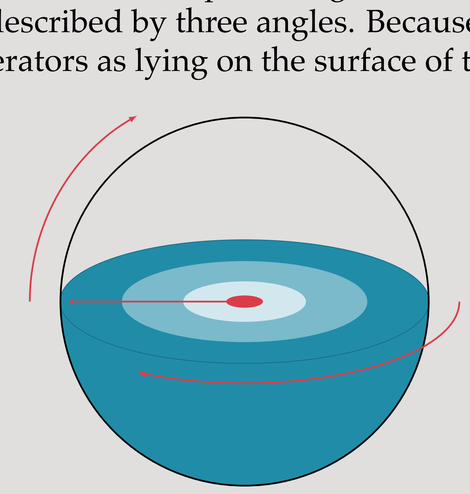


Figure 7: A pictorial representation of the 3-sphere: the upper half has been removed for clarity. The three arrows represent the degrees of freedom along the 3-sphere's surface, the points of which have been projected to the interior of a 2-sphere. (The arrow along the traditional `radius', if extended, gives degeneracy). Despite implication there is no preferred direction along which to move across the 3-sphere.

Arguments & Degrees of Freedom in the Test Organism

For the adaptive algorithm process described, each organism, denoted Θ, is given a set of arguments which define it and serve as its characteristics. For our purposes, these can be further divided into *constants* and *mutables*.
Among the constants, we have H, a set Hamiltonian that is easy to generate, and is used in every free evolution between LO. It does not change in each simulation, across all organisms, and across all generations. We also have N, which defines the number of LO-FE pairs for all organisms in a given simulation. None of these variables is subject to the mutation protocol that defines adaptive algorithms.

$$\underbrace{(A_1 \otimes B_1)e^{-iHt_1} \cdots (A_N \otimes B_N)e^{-iHt_N}}_{N} \tag{4}$$

The mutable data contains all of the sets. This includes the set of A_i, which are unitary operators on the first particle, the set of B_i, which are unitary operators on the second particle, and the set of t_i, which are the discrete durations for each FE, and have the constraint of summing to a particular value.
In any effective adaptive algorithm, the structure of the search space must be known. This allows both a functioning and efficient algorithm. In this scenario, the dimension of the space depends on the step number N in a linear fashion. We begin with a local operation, A, which is of the form

$$\beta \cdot \sigma = \begin{pmatrix} \beta_0 + \beta_3 & \beta_1 - i\beta_2 \\ \beta_1 + i\beta_2 & \beta_0 - \beta_3 \end{pmatrix} \tag{5}$$

This appears to have four degrees of freedom, one for each element of β. However, there is an implicit restraint given the unitarity of the LO, we may recognize this as an exponentiation of a linear combination of the pauli-matrices.

$$A^\dagger A = \mathbb{1} \implies A = \mathbb{1}\cos|\beta'| + (\beta' \cdot \sigma)\sin|\beta'| \tag{6}$$

And we recognize this form as being equivalent to the specification of a single point by a new vector β' on the three sphere. Thus A is also equivalent to

$$exp\{\beta' \cdot \sigma\} = exp\{\beta'_1\sigma_1 + \beta'_2\sigma_2 + \beta'_3\sigma_3\} \tag{7}$$

This gives six degrees of freedom for each pair of local operations. We also note that given variability in the discrete FE times, we would be given $N-1$ more degrees of freedom (the restriction is due to their sum being defined). All considered, we are searching a $6N$-dimensional space for every N-family.

Conclusion

We have identified various aspects of using adaptive algorithms that are beneficial and detrimental in determining local operation sets to simulate an arbitrary gate. Although Our results have consistently agreed mathematically, with fairly large fidelity, the run-time grows exponentially with the number of organisms, generations, and gene number. This can only be countered well by running this program with multiple processors (i.e. in parallel, which this algorithm is well-suited for) [5]. Moreover, a lot of lag in the code is derived from matrix multiplication, which is a daunting task to optimize. Perhaps, using a more effective algorithm will improve the run-time efficiency of the code as well. We can however partially ease matrix multiplication by taking advantage of particular expansions of matrix exponentials, which can be truncated depending upon our desired accuracy. Using commutation relations and Lie algebras may immensely help with the reduction of run-time as well (this involves both engineering the heuristic and the initial populations). The methods forming the basis of this scheme are naive, and base little of their workings off of the defining tricks of linear algebra and group theory. There are properties to be exploited, but we chose to produce a functional process before an optimal one. We also wish to represent better ways of representing the resulting operators, to give more intuition about their structure.
Although we compromise time for efficiency, our efforts in implementing adaptive algorithms to solve this quantum computation problem has yielded high success in producing a viable solution sets. With complex systems, as are necessary to produce non-trivial quantum computations, analytical methods for generating the correct unitary evolution operator are simply not feasible. With the proper improvements, non-deterministic methods such as the one here are vital to the success of quantum computation.

References

[1] Diederik Aerts. The extended bloch representation of quantum mechanics and the hidden-measurement solution to the measurement problem. *Annals of Physics*, 351:975--1025, December 2014.
[2] C.H. Bennett. Optimal simulation of two-qubit hamiltonians using general local operations. 2008.
[3] Michael J. Bremner. Practical scheme for quantum computation with any two-qubit entangling gate. *Physical Review Letters*, 89(24), 2002.
[4] Henry L. Haselgrove. Practicality of time-optimal two-qubit hamiltonian simulation. *Physical Review*, 2003.
[5] A.J. Umbarkar. Multithreaded parallel dual population genetic algorithm (mpdpga) for unconstrained function optimizations on multi-core system. *Applied Mathematics and Computation*, 243:936--949, September 2014.
[6] Thomas Weise. *Global Optimization Algorithms: Theory and Application*. 2009.