Implementation of a Quantum Trajectory Algorithm by Diagonalization of the Effective Hamiltonian

Douglas M. McNally* and James P. Clemens†

Department of Physics, Miami University, Oxford, Ohio, 45056

(Dated: July 8, 2014)

Abstract

A quantum trajectory simulation is a Monte Carlo method for investigating open quantum systems. The fundamental object in a quantum trajectory is a conditional wavefunction which is evolved in time according the the Schrodinger equation with a non-Hermitian effective Hamiltonian and punctuated by stochastic quantum jumps generated by the coupling between the system and the environment. By averaging over many independent conditional wavefunctions one can reproduce the results calculated from the master equation for the time evolution of the reduced density operator for the system. It is often computationally preferable to use the Monte Carlo approach instead of directly solving the master equation particularly for problems with a large Hilbert space. On the other hand the Monte Carlo approach is useful to investigate conditional dynamics which would be averaged over by the master equation even for small problems. Existing implementations of the quantum trajectories algorithm, such as `mcsolve` in the popular quantum optics package The Quantum Toolbox in Python (QuTiP) [Comput. Phys. Commun., **184**, 1234 (2013)], depend on high order differential equation solvers to obtain the time evolution in between the quantum jumps. Herein an alternate method which utilizes matrix diagonalization to simplify the time evolution is presented and implemented in Python and Fortran. This also allows the determination of the time of the next quantum jump using only a simple zero-finding technique and eliminates the need for small time steps to maintain high precision in the time evolution of the wavefunction. Agreement with existing implementations and master equation solutions is shown and performance of the different algorithms is compared. We find that the approach described here is faster for problems with a small Hilbert space dimension.

PACS numbers: 02.70.-c, 03.65.Yz

* mcnalldm@miamioh.edu

† clemenj2@miamioh.edu

I. INTRODUCTION

Describing real experiments involving quantum systems requires a theory accounting for open quantum systems which interact with their environment. It is often difficult or impossible to analytically solve the equations governing these systems and thus numerical methods for investigating open quantum systems are of interest. One formulation of the time evolution of open quantum systems in the Born-Markov approximation leads to a master equation in the Lindblad form [1]:

$$\dot{\rho} = -\frac{i}{\hbar}[H_{\text{sys}}, \rho] + \sum_i (2\hat{C}_i \rho \hat{C}_i^\dagger - \hat{C}_i^\dagger \hat{C}_i \rho - \rho \hat{C}_i^\dagger \hat{C}_i) \tag{1}$$

where H_{sys} is the system Hamiltonian and $\hat{C}_i$ are "collapse" operators that account for the coupling between the environment and the system. It is possible to solve this equation directly numerically, but it requires using the density operator ρ which has D^2 entries in its matrix representation where D is the Hilbert space dimensionality. This is not ideal since it means the amount of data one has to keep track of scales as the square of the dimensionality of the Hilbert space. Quantum trajectory simulations are an alternate, but equivalent method of obtaining the time evolution of the density operator. One treats the problem with a Monte-Carlo approach, utilizing ensemble averaging over many independent realizations of the system in order to arrive at a result that is on average equivalent to the master equation [1–4].

Interestingly, quantum trajectories can actually reveal information about a system that is hidden to the master equation solution. Since the master equation describes the average behavior of an ensemble of systems, the dynamics of the constituents of the ensemble are not reported. By contrast the quantum trajectories approach produces a set of simulated measurement records and a set of wavefunctions conditioned on the corresponding measurement records [1, 2].

Each individual trajectory starts with the same initial state and is evolved forward in time according to the Schrödinger equation:

$$i\hbar \, |\dot{\psi}\rangle = H_{\text{eff}} \, |\psi\rangle \tag{2}$$

with a non-Hermitian effective Hamiltonian

$$H_{\text{eff}} = H_{\text{sys}} - i\hbar \sum_i \hat{C}_i^\dagger \hat{C}_i \tag{3}$$

where again, $\hat{C}_i$ are so-called "collapse operators" which represent the interaction between the system and its environment. Note that the anti-Hermitian part of the effective Hamiltonian causes the norm of the wavefunction to monotonically decrease in time. The decreasing norm can be used to determine the time of the next stochastic quantum jump with the correct statistics such that the average over many wavefunctions reproduces the results of the master equation. At each time step of length Δt, a decision is made to either evolve the current state up to the next time step, or apply a collapse operator with probability

$$P_i = \langle C_i^\dagger C_i \rangle \Delta t \tag{4}$$

and renormalize the state. Alternatively, assuming the wavefunction is normalized at time t, the probability of evolving without a collapse for a time τ is

$$P_\tau = \langle \overline{\psi(t+\tau)} | \overline{\psi(t+\tau)} \rangle \tag{5}$$

where the overline indicates we are using the unnormalized wavefunction. After determining the time of the next quantum jump the particular collapse operator to be applied is chosen according to the relative probability of each, given by

$$P_i = \frac{\langle \psi(t) | \hat{C}_i^\dagger \hat{C}_i | \psi(t) \rangle}{\sum_j \langle \psi(t) | \hat{C}_j^\dagger \hat{C}_j | \psi(t) \rangle}. \tag{6}$$

Existing implementations [5–7], of the quantum trajectory algorithm use numerical integration of the Schrödinger equation in order to obtain the time evolution of the state vector. This could be realized by Euler integration as

$$|\overline{\psi(t+\Delta t)}\rangle = (1 - \frac{i}{\hbar} H_{\text{eff}} \Delta t) \, |\overline{\psi(t)}\rangle \, . \tag{7}$$

Thus it requires small time-steps for accuracy (which can be mitigated with higher-order ODE algorithms), and calculating the norm of the state at each time-step.

II. IMPLEMENTATION DETAILS

Note that the solution to the Schrödinger equation in closed form could be written as

$$|\overline{\psi(t)}\rangle = \mathrm{e}^{-\frac{i}{\hbar} H_{\text{eff}} t} \, |\psi(0)\rangle \, . \tag{8}$$

Recall that the non-unitarity of the time evolution results from the non-Hermitian effective Hamiltonian. While this expression is not useful for numerical computation in general, in

the special case where H_{eff} can be written as a diagonal matrix then the expression forms the basis for an alternative quantum trajectory algorithm. For a matrix A, if we can find eigenvectors of A then there exists an invertible similarity transformation matrix S whose columns are the right eigenvectors of A, such that $S^{-1}AS = \Lambda$ where Λ is a diagonal matrix with the eigenvalues of A on its diagonal. Extending this to the notion of matrix exponentiation which has meaning through its Taylor series representation, it can be shown [8] that

$$\mathrm{e}^{At} = S\mathrm{e}^{\Lambda t}S^{-1}. \tag{9}$$

This of course has a consequence for the solution to the Schrödinger equation, namely

$$|\overline{\psi(t)}\rangle = S\mathrm{e}^{\Lambda t}S^{-1}\,|\psi(0)\rangle \tag{10}$$

where here $A = -\frac{i}{\hbar}H_{\text{eff}}$, and since the exponential of a diagonal matrix is just the exponential of each of the diagonal elements, this greatly simplifies the time evolution. It also means that it is possible to calculate the state at a later time in a single operation and thus find the times at which "jumps" (applications of collapse operators) will occur in advance using a zero-finding algorithm

$$\langle\overline{\psi(t)}|\overline{\psi(t)}\rangle - r = 0 \tag{11}$$

where r is a random number in the range $[0, 1]$, and a collapse occurs when the norm of the state vector falls below r. This eliminates any backtracking that might be necessary in a numerical integration approach in the case that the timestep being used overshoots the time of a jump. It should be noted that it is unclear if the effective Hamiltonian is always diagonalizable; however this approach has utility even if it is not, because it is known to be for many problems of interest. Another caveat is this method will only work for problems that have no explicit time dependence in the system Hamiltonian or the collapse operators, as this would cause the similarity transformation matrices to be time dependent.

To illustrate the differences that the matrix diagonalization technique introduces to the algorithm, consider a pseudocode outline of the procedure in the two cases. These algorithms assume a discrete set of increasing times at which to compute the state is provided, i.e. `tlist` $= [0, t_1, t_2, \cdots, t_n]$.

a. Quantum trajectories using numerical integration

$t \leftarrow t_0$

while $t < t_n$ **do**
  evolve $|\overline{\psi(t)}\rangle \rightarrow |\overline{\psi(t_{i+1})}\rangle$
  compute $p(t_{i+1}) = \langle\overline{\psi(t_{i+1})}|\overline{\psi(t_{i+1})}\rangle$
  if $p(t_{i+1}) > r$ **then**
    calculate expectation values
    $t \leftarrow t_{i+1}$
    increment i
  else
    backtrack to t where $p(t) = r$
    apply collapse $|\overline{\psi(t)}\rangle = \hat{C}\,|\overline{\psi(t)}\rangle$
    normalize $|\psi(t)\rangle$
  end if
end while

Note that additional calculations may be needed in the case that the time of a jump lies somewhere between t_i, t_{i+1}.

b. Quantum trajectories using matrix diagonalization

$t \leftarrow t_0$
while $t < t_n$ **do**
  calculate time of next jump $= t_{\text{jump}}$ with $\langle\overline{\psi(t_{\text{jump}})}|\overline{\psi(t_{\text{jump}})}\rangle - r = 0$[1]
  while $t_i < t_{\text{jump}}$ **do**
    evolve to t_i
    calculate expectation values
    increment i
  end while
  evolve to t_{jump}
  apply collapse $|\overline{\psi(t_{\text{jump}})}\rangle = \hat{C}\,|\overline{\psi(t_{\text{jump}})}\rangle$
  normalize $|\psi(t_{\text{jump}})\rangle$
end while

Both procedures run until a maximum time t_{n}, and at each time in `tlist` various expectation values would be calculated.

[1] Brent's method was used in the implementation discusssed here.

When using quantum trajectories, the user typically specifies some operators $\hat{O}$ corresponding to observables for which they wish to know the expectation value at a discrete set of times. These expectation values are calculated as always, $\langle \hat{O} \rangle (t) = \langle \psi(t) | \hat{O} | \psi(t) \rangle$ at each t that is requested. If the list of times are equally spaced by an amount Δt, i.e. $t_i = i\Delta t$, then the state vector at a time t_i can be calculated as

$$|\overline{\psi(t_i)}\rangle = [S \mathrm{e}^{\Lambda \Delta t} S^{-1}]^i |\psi(0)\rangle \tag{12}$$

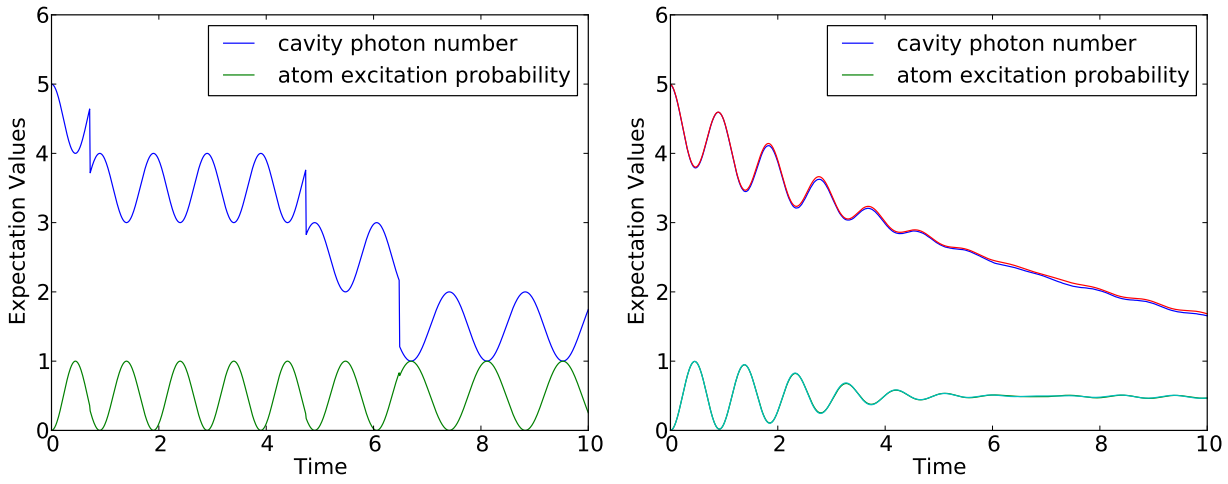
and note that the linear transformation $S\mathrm{e}^{\Lambda \Delta t} S^{-1}$ is constant and therefore only needs to be calculated once. When jumps occur the time evolution goes up to t_{jump} which is generally not a multiple of Δt and so there will be an occasional need to compute the linear transformation $S\mathrm{e}^{\Lambda t} S^{-1}$ with $t \neq \Delta t$ but often the vast majority of time displacements are by Δt. Therefore this can provide a substantial improvement in the total amount of matrix multiplications that need to be performed.

III. COMPARISON WITH MASTER EQUATION SOLUTIONS

Figure 1 helps to illustrate how the quantum trajectories algorithm works in practice. A simulation was run for a two-level atom coupled to an optical cavity that initially contained 5 photons. The effective quantum trajectory Hamiltonian for this is

$$H_{\mathrm{eff}} = \hbar\omega(a^\dagger a + \sigma_+ \sigma_-) + \hbar g(\sigma_- a^\dagger + \sigma_+ a) - i\hbar\kappa a^\dagger a \tag{13}$$

where ω is the atom and cavity resonance frequency, g is the coupling strength, and κ is the photon leakage rate. Expectation values are plotted against time; specifically Fig. 1(a) shows the expectation values for a single trajectory. There are clear discontinuous jumps in the plot of the expectation value of the number of photons in the cavity. These jumps correspond to a photon leaving the cavity, or in terms of the algorithm, they correspond to a collapse operator being applied to the state. Fig. 1(b) shows the result of the ensemble average of 5000 trajectories like those in Fig. 1(a). Also plotted is the solution from the master equation, and clearly there is excellent agreement between the two—they are nearly coincident and the agreement will increase as more trajectories are averaged. These plots were generated using the matrix diagonalization techniques described above, and therefore provide some evidence that this alternate algorithm does function as claimed.



(a) (b)

FIG. 1: (a) A single trajectory for a two-level atom coupled to a leaky cavity initially containing 5 photons. The discrete jumps in the cavity photon number correspond to a collapse operator being applied to the state—specifically a photon leaving the cavity. (b) 5000 trajectories like those in (a) averaged together and plotted along with the master equation solutions (red and light green) for comparison. Here $\hbar\omega = 2\pi$, $\hbar g = \frac{\pi}{2}$, and $\hbar\kappa = \frac{1}{20}$.

Fig. 2 shows the results for a two-level system (initially in the ground state) being driven on resonance by an external field. The effective quantum trajectory Hamiltonian for this problem is

$$H_{\text{eff}} = \hbar\Omega\sigma_x - i\hbar\gamma\sigma_+\sigma_- \tag{14}$$

where Ω is the Rabi frequency and γ is the spontaneous emission rate. Again there is clearly excellent agreement with the solution from the master equation, showing again that everything is working as intended.

IV. COMPARISON OF PERFORMANCE

For the reasons discussed previously, there is hope that this alternate algorithm will be faster for some problems, or perhaps in general. This algorithm has been implemented in a form that could be suitable for inclusion in the Quantum Toolbox in Python (QuTiP) [6, 7]. To that end, a version was developed purely in Python. QuTiP presently has an optional

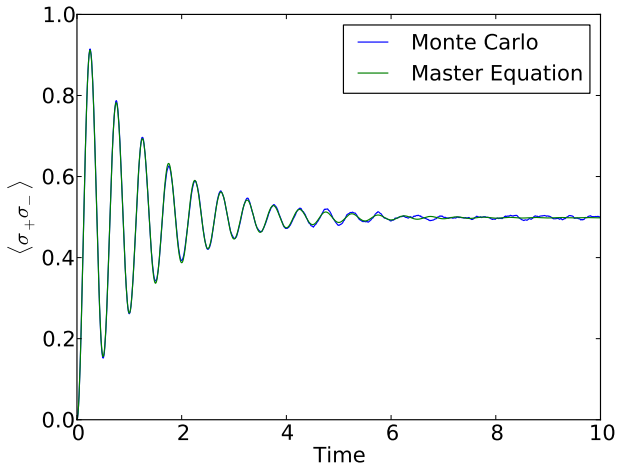
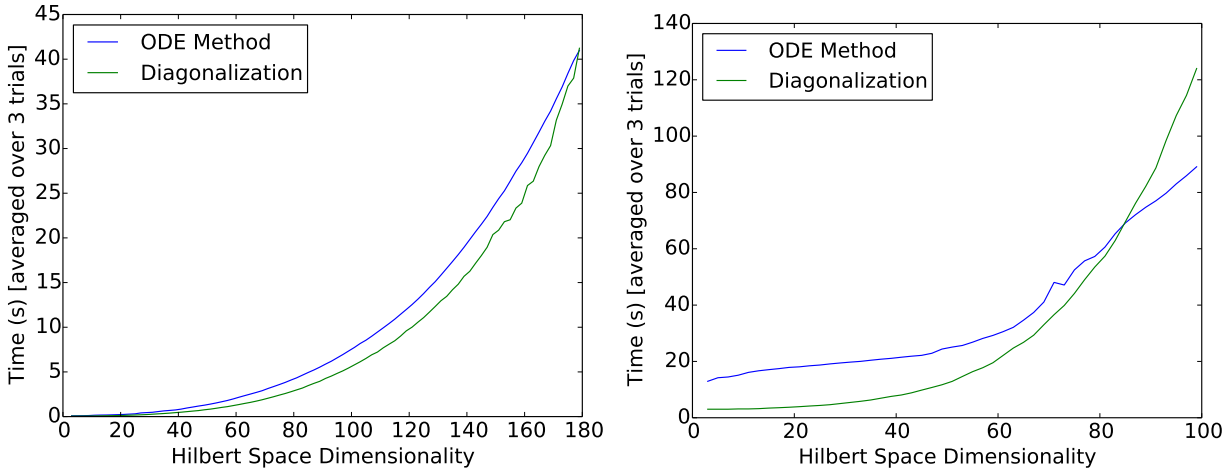


FIG. 2: A two-level atom being driven on resonance by an external field and allowed to spontaneously emit (resonance florescence). 5000 trajectories plotted alongside the master equation solution (green) for comparison. Here $\hbar\Omega = 2\pi$ and $\hbar\gamma = \frac{1}{2}$.

Monte-Carlo solver with a Fortran 90 backend using F2PY [9]. The algorithm described here was also implemented in Fortran 90 by modifying the existing QuTiP code for that implementation. The performance will depend most strongly on the language used with the Fortran implementation substantially faster than the Python implementation. Aside from the implementation language the most important predictor of performance is the dimension of the Hilbert space of the problem.

For problems with small Hilbert spaces, this method can be considerably faster. For the example shown in Fig. 2 the algorithm described is roughly five times faster than the Fortran implementation of quantum trajectories in QuTiP. However this represents the smallest possible Hilbert space that could be encountered in a nontrivial quantum mechanical calculation. It is not clear that this will scale to large Hilbert spaces, which presents a substantial limitation. Small problems can already be solved quickly with a master equation approach; however, quantum trajectories are still useful for investigating conditional dynamics in small problems.

An investigation into the crossover in the performance of the two algorithms is presented in Fig. 3. A driven Dicke model [10] was used as the example problem for convenience, since it exhibits linear scaling of the size of the Hilbert space with the number of atoms N. We solve the master equation for N atoms collectively coupled to the electromagnetic field and



(a) (b)

FIG. 3: Demonstration of the performance crossover in terms of the size of the Hilbert space between the new (diagonalization) method and the existing (ODE) method. A driven Dicke Model was used. (a) Shows performance for Fortran 90 implementations, (b) is for pure Python implementations. The simulation was run on Ubuntu 14.04 LTS with an AMD FX-8350 CPU clocked at 4.0GHz.

driven collectively by an external laser,

$$\dot{\rho} = \hbar Y \left(J_+ + J_-\right) + \frac{\gamma}{2}\left(2J_-\rho J_+ - J_+J_-\rho - \rho J_+J_-\right) \tag{15}$$

where $J_- = \sum_{i=1}^{N} \sigma_{i-}$, σ_{i-} is the Pauli lowering operator for the ith atom, Y is the strength of the driving field, and γ is the spontaneous emission rate. In Fig. 3 it can be seen that the point at which the new method starts to become slower than the existing algorithm is around a Hilbert space dimensionality of $D = 180$ for the Fortran 90 implementation and $D = 85$ for Python. Unfortunately this often is not even large enough to warrant the use of quantum trajectories in the first place (usually for D on the order of several hundred). There are some exceptions of course for situations where information about the conditional dynamics of a system is desired; this information cannot be easily obtained from a master equation solution. It is worth noting however that the crossover point does depend on the character of the problem in question, and even for the problem used here it had a non-negligible dependence on the amount of driving. In addition, since the calculations rely heavily on matrix multiplication it was found that the particular version of the Basic Linear Algebra Subprograms (BLAS) called by Fortran 90 substantially influenced performance,

and the version that offered the best performance on the hardware used was OpenBLAS [11]. The likely reason for this method being slower for large problems is that the similarity transformation matrices are generally dense. By contrast, the Hamiltonian and collapse operators are generally extremely sparse which can give the numerical integration approach an advantage when it comes to the actual calculations even though it ostensibly has more steps at a high level. This means that the inevitability of using dense matrices in the calculations ultimately seems to be too high of a computational cost and wins out over whatever savings are gained by eliminating the need to compute the norm at each time step and eliminating backtracking as the dimension of the Hilbert space increases. On the other hand there may be problems of interest where the enhanced speed of the new algorithm is useful for investigating conditional dynamics.

V. CONCLUSION

An alternate method for finding the time evolution of a state vector by computing the exponential of the diagonal time-independent effective Hamiltonian in quantum trajectory simulations was presented. Evidence that this method is valid was given by computing solutions for several problems and comparing this with the known results. Computational performance was then compared between Fortan 90 implementations of this new algorithm and the `mcsolve_f90` method from QuTiP [7]. For problems with small Hilbert space dimensionality, the matrix diagonalization technique is definitively faster when tested on several consumer-grade CPUs. When it comes to problems with larger Hilbert spaces the performance benefit drops off and at some point the numerical integration approach becomes favorable again. In order to quantify this crossover, a problem whose Hilbert space scales linearly with the number of atoms was chosen and the results demonstrated that there is a clear range over which this new approach is useful. Which method is preferable will ultimately depend upon the particular problem in question, but nevertheless there is strong evidence that at least some circumstances using the technique described is beneficial in terms

of computational time.

---

[1] H. J. Carmichael, *Statistical Methods in Quantum Optics 2: Non-Classical Fields* (Springer-Verlag, 2008).

[2] H. J. Carmichael, *An Open Systems Approach to Quantum Optics*, Lecture Notes in Physics, New Series: Monographs, Vol. m18 (Springer, Berlin, 1993).

[3] K. Mølmer and Y. Castin, Quantum and Semiclassical Optics: Journal of the European Optical Society Part B **8**, 49 (1996).

[4] A. J. Daley, "Quantum trajectories and open many-body quantum systems," (2014), arXiv:1405.6694 [quant-ph].

[5] S. M. Tan, J. Opt. B: Quantum Semiclass. Opt. **1**, 424 (1999).

[6] J. R. Johansson, P. D. Nation, and F. Nori, Comp. Phys. Comm. **183**, 1760 (2012).

[7] J. R. Johansson, P. D. Nation, and F. Nori, Comp. Phys. Comm. **184**, 1234 (2013).

[8] C. Moler and C. Van Loan, SIAM Review **45**, 3 (2003), http://epubs.siam.org/doi/pdf/10.1137/S00361445024180.

[9] P. Peterson, International Journal of Computational Science and Engineering **4**, 296 (2009).

[10] R. H. Dicke, Phys. Rev. **93**, 99 (1954).

[11] Z. Xianyi, W. Qian, and W. Saar, "OpenBLAS," `http://xianyi.github.com/OpenBLAS` (2013).