Testing quantum state engineering protocols via LIQ$Ui|\rangle$ simulations

András Gilyén[1,2]*
[1] *Centrum Wiskunde & Informatica, Amsterdam, Netherlands and*
[2] *University of Amsterdam, Netherlands*

In this report I show how to implement various iterative state preparation protocols in LIQ$Ui|\rangle$. I start with a post-selective protocol which implements non-linear transformations of the quantum state of qubits. The family of the implemented circuits comes from my recently published article [1], where we show how to implement Mandelbrot maps on a bunch of qubits. The resulting circuits make it possible to draw the Mandelbrot set using (simulated) measurement statistics of its output.

In the second part I test the performance of a recently proposed quantum algorithm which attempts constructing the ground state of a frustration free local Hamiltonians [3, 7]. The algorithm is proven to be effective in a restricted case only considering commuting Hamiltonians [7]. The major open question for which I tried and gain insight is whether the algorithm remains effective for non-commuting Hamiltonians. The simulation results suggest that the expected runtime of the algorithm depends on the eigenvalue gap of the Hamiltonian and may only be effective if its eigenvalue gap is not too small.

Introduction

Preparing specific quantum states can be a challenging task, and there is a lot of work around this topic in quantum computation and information theory. Although we can build efficient circuits that implement various non-trivial transformation of quantum states, it can be shown using a counting argument that the majority of quantum states are hard to prepare. Still there is a large space to explore and researches found some nice new protocols over the recent years with possible algorithmic implications. In fact some of the new quantum algorithms transform a computational problem to a state engineering task to harness the power of quantum entanglement. Probably the most prominent example is the HHL[9] algorithm which solves some special kind of linear equations cast into a quantum state preparations framework. This algorithm does the following trick: instead of solving the linear equations $Ax = b$ directly it transforms an input quantum state proportional to $x \sim |x\rangle$ to an output state $|b\rangle \sim b$. If we can extract the features of x we are interested in from $|x\rangle$ we can make use of the quantum speed-up which may be exponentially large in the number of variables. In order to build new quantum algorithms we need more building blocks. Some new quantum algorithms use insights coming from quantum state engineering, so it does worth looking at prospective protocols along this line. In this report we examine two different kinds of state transformation/preparation protocols.

The first half of this report consists of simulation results of a post-selective protocol which transforms quantum states in a non-linear way. The goals of this part were to learn how to use LIQ$Ui|\rangle$, verify that the circuits presented in my recent paper are correct and obtain the first "quantum computer produced" images of the Mandelbrot set.

Some of the maps in [1] separate initially close quantum states exponentially fast, giving rise to a "Schrödinger-microscope" [2] like effect. Although this kind of effect was proposed as a possible tool for achieving quantum speed-up [2], our "Quantum Magnification Bound" [1] suggest that we may not gain significant algorithmic speed-up this way. Still it worth exploring the resulting highly non-trivial maps even just for their beauty.

The goal of the second part of my project was to gain some insight on the performance of the quantum Moser-Tardos algorithm in the general non-commutative case. This is an interesting algorithm which attempts to prepare a ground state of a frustration free local Hamiltonian. (A local Hamiltonian is one which is a sum of local terms each acting only on a few qubits. Such a Hamiltonian is called frustration free if there is a global ground state which is also a ground state of all local terms.)

I could recently discuss issues about this open problem with almost all the researchers from my field interested in this question, but none of them were aware of any numerical simulations done. It was not even clear whether one should try to refute or prove efficiency in the non-commutative case, and I thought numerical trials may give a hint on which direction to choose.

What is known that dissipative ground state preparation actually converges to the ground state [3] but the convergence can be very slow. However one might be able to find a ground state of a local Hamiltonian efficiently if it is guaranteed to exists by the Lovász[4, 6] or Shearer[5, 8] criterion. Fast convergence of the quantum Moser-Tardos algorithm has been proven in a restrictive setting where only commuting local Hamiltonians are allowed [7]. The big open question here is whether the algorithm still converges with a decent speed when we allow non-commuting pairs.

*Electronic address: gilyen@cwi.nl

Simulating a "Schrödinger-microscope" and Mandelbrot maps

The basic idea of the post-selective maps simulated here comes from entanglement distillation protocols. We start with two identical copies of qubits and apply a CNOT gate to them followed by a measurement. We only keep the other qubit if the measurement resulted in 0. After post-selection the state of the remaining qubit can be described by a non-linear transformation. Later we also apply some unitary transformation to it, see Figure 1. To understand this map first omit the extra unitary step, and see how $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ transforms:
$|\psi\rangle \otimes |\psi\rangle = \alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|10\rangle + \beta^2|11\rangle \overset{CNOT}{\longrightarrow}$
$\alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|01\rangle + \beta^2|10\rangle \overset{p-s}{\longrightarrow} \sim \alpha^2|0\rangle + \beta^2|1\rangle$
So the new state vector is proportional to $\alpha^2|0\rangle + \beta^2|1\rangle$. In order to handle the arising non-linear map easily, we introduce another parametrisation of the qubits states: $z = \frac{\alpha}{\beta} \in \mathbb{C} \cup \{\infty\}$. So the above transformation can be described by the map $z \to z^2$, and the post-selective map F arising after the unitary gate on Figure 1 becomes $z \to \frac{z^2+i}{iz^2+1}$. According to [1] the iteration of the map F results in a kind of "Schrödinger microscope" which separate initially close quantum states exponentially fast. But can we see the traces of this behaviour using LIQ$Ui|\rangle$? The results are illustrated on Figure 2 – observe the rapid alternation of the pattern eventually distinguishing even very close initial states.

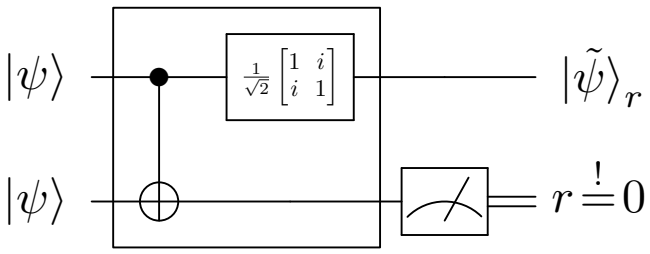
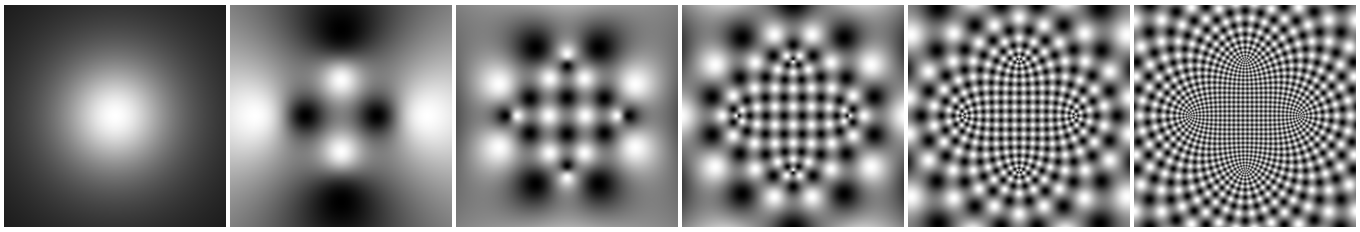


Figure 1: A post-selective circuit F accepting only 0 measurement outcome on the second qubit.



(a) Prob1($|\psi_z\rangle$) (b) Prob1($F^{\circ 2}|\psi_z\rangle$) (c) Prob1($F^{\circ 4}|\psi_z\rangle$) (d) Prob1($F^{\circ 6}|\psi_z\rangle$) (e) Prob1($F^{\circ 8}|\psi_z\rangle$) (f) Prob1($F^{\circ 10}|\psi_z\rangle$)

Figure 2: **Simulation of a "Schrödinger microscope" by LIQ$Ui|\rangle$** The subfigures cover the range $z \in [-2;2] \times [-2i;2i] \subset \mathbb{C}$ and the points correspond to initial quantum states $|\psi_z\rangle$ proportional to $z|0\rangle + 1|1\rangle$. Each pixel is coloured according to the probability of measuring $|1\rangle$ after applying the post-selective circuit F iterated n times. (Brighter means higher chance.) For spatial reasons only iterates after even steps are plotted.

The above scheme can be generalised and one can devise a circuit for the Mandelbrot maps $z \to z^2 + c$ as well. The circuit M_c from [1] shown in Figure 3 uses only controlled one qubit gates and implements all Mandelbrot maps.

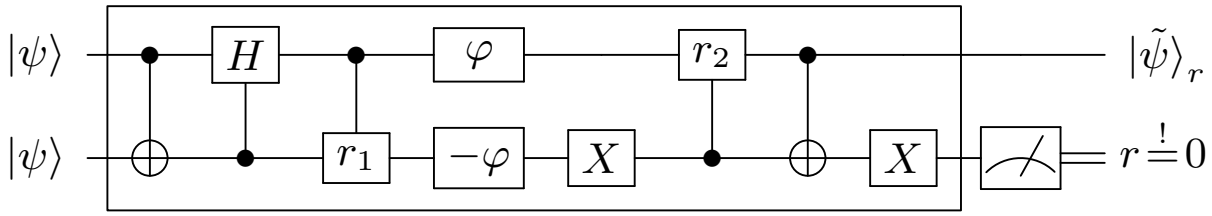


Figure 3: **A quantum circuit implementing Mandelbrot maps.** The controlled gate labelled by the real number r_j is essentially a rotation composed with a Pauli-Z-gate $\frac{1}{\sqrt{1+r_j^2}}\begin{bmatrix}1 & r_j\\ r_j & -1\end{bmatrix}$ and the gate labelled by $\pm\varphi$ is a phase gate $\begin{bmatrix}1 & 0\\ 0 & e^{\pm i\varphi}\end{bmatrix}$, while H and X stand for the Hadamard and Pauli-X gates correspondingly. If we set $r_1 = 1/r_2$, $r_2 = |c| \cdot \sqrt{(1+\sqrt{1+4/|c|^2})/2}$ and $e^{i\varphi} = \frac{c}{|c|}$, then the resulting map is $z^2 + c$, provided that we accept only the 0 measurement outcome on the second qubit. A possible LIQ$Ui|\rangle$ implementation is shown below.

The following pieces of code show my implementation of the post-selection and iteration protocol as well as the parametrised Mandelbrot circuit M_c. Note that for performance reasons I allow LIQ$Ui|\rangle$ to cheat and clone quantum states, otherwise we would need to prepare an exponential amount of identical initial qubits copies. (For spatial reasons I did not include my straightforward custom gate definitions used as the building blocks of M_c.)

```
let rec Iter (ket:Ket) time gate =
    if time < 0 then invalidArg "time"
        "Can not iterate function backwards!"
    if time = 0 then
        ket
    else
        let qs = ket.Qubits
        gate qs
        M[qs.[1]]
        if qs.[1].Bit.v = 0 then
            qs.[1].ReAnimate qs.[0].State;
            qs.[2].StateSet qs.[0].State
            Iter ket (time - 1) gate
        else
            qs.[1].ReAnimate qs.[2].State;
            qs.[0].StateSet qs.[2].State
            Iter ket time gate
```

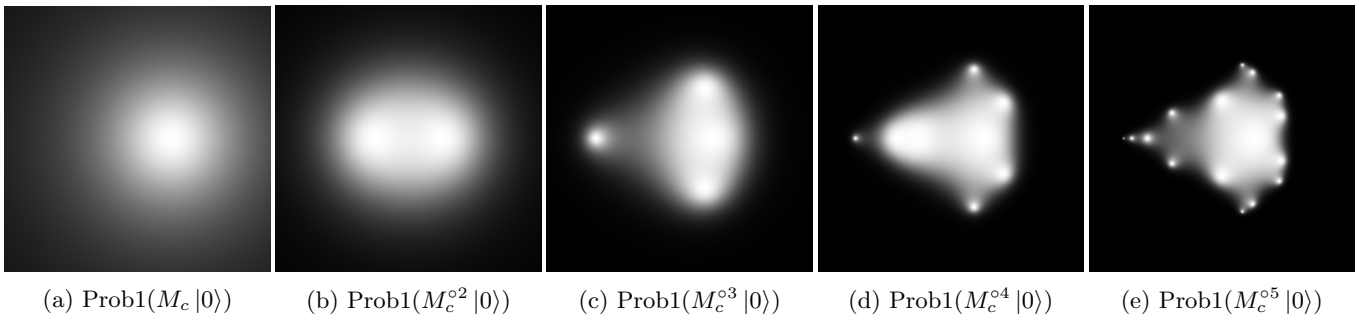
(a) LIQ$Ui|\rangle$ simulation of iteration & post-selection.

```
let MandelCircuit (c:Complex) (qs:Qubits) =
    if c.r = 0. && c.i = 0. then CNOT qs
    else
        let q0,q1 = qs.[0],qs.[1]
        let phi = Complex.arg c
        let r2 = (Complex.abs c)*sqrt((1. + sqrt(
            1. + 4./c.MCC)) / 2.)
        let r1 = 1./r2
        CNOT [q0;q1]
        (CRotZ 1.) [q1;q0]
        (CRotZ r1) [q0;q1]
        (phaseZ phi) [q0]
        (phaseZ -phi) [q1]
        X [q1]
        (CRotZ r2) [q1;q0]
        CNOT [q0;q1]
        X [q1]
```

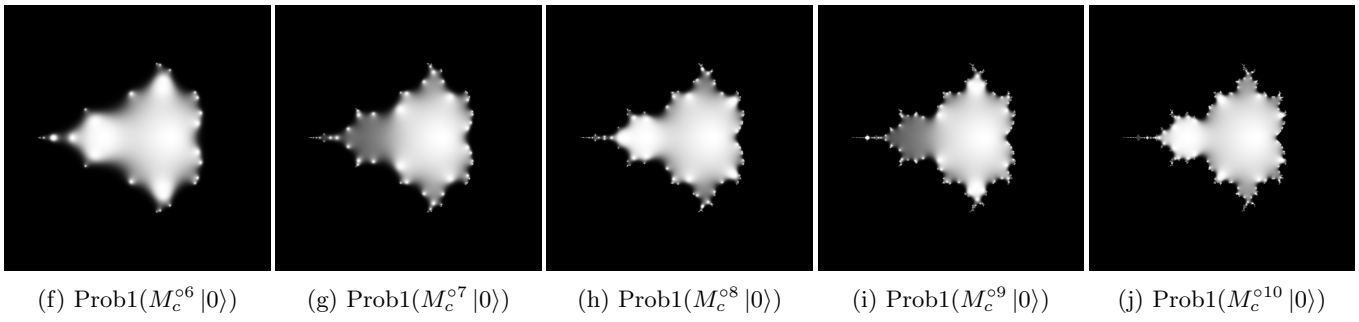
(b) LIQ$Ui|\rangle$ implementation of the circuit M_c.

(The above circuit M_c is not optimised, but due to the small sizes the simulation was still fast enough for my purposes.)

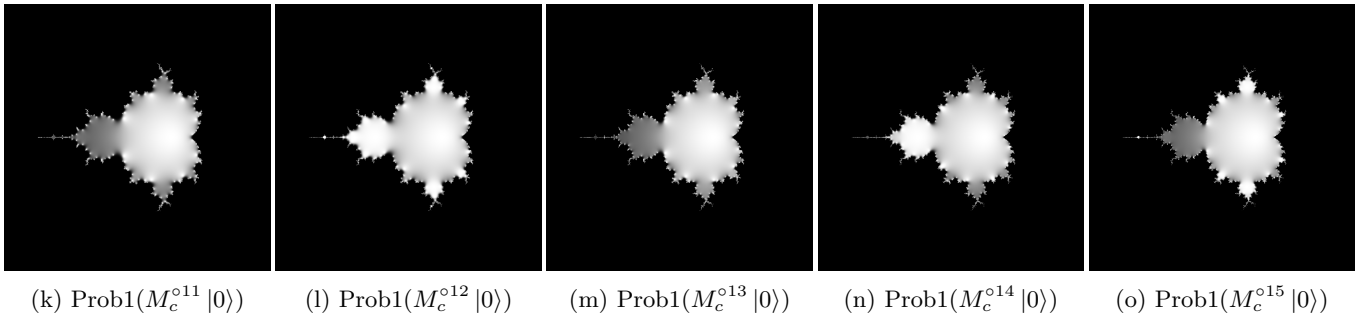
Using the above building blocks and a little bit of additional handling code we can simulate measurement probabilities of Mandelbrot circuits. The results shown in Figure 5 confirm that both the M_c circuits and their LIQ$Ui|\rangle$ implementation match the predictions of the theory.



(a) Prob1($M_c\,|0\rangle$) (b) Prob1($M_c^{\circ 2}\,|0\rangle$) (c) Prob1($M_c^{\circ 3}\,|0\rangle$) (d) Prob1($M_c^{\circ 4}\,|0\rangle$) (e) Prob1($M_c^{\circ 5}\,|0\rangle$)



(f) Prob1($M_c^{\circ 6}\,|0\rangle$) (g) Prob1($M_c^{\circ 7}\,|0\rangle$) (h) Prob1($M_c^{\circ 8}\,|0\rangle$) (i) Prob1($M_c^{\circ 9}\,|0\rangle$) (j) Prob1($M_c^{\circ 10}\,|0\rangle$)



(k) Prob1($M_c^{\circ 11}\,|0\rangle$) (l) Prob1($M_c^{\circ 12}\,|0\rangle$) (m) Prob1($M_c^{\circ 13}\,|0\rangle$) (n) Prob1($M_c^{\circ 14}\,|0\rangle$) (o) Prob1($M_c^{\circ 15}\,|0\rangle$)

Figure 5: **The Mandelbrot set arising from measurement statistics of a family of quantum circuits.** The subfigures cover the range $c \in [-2.5, 1.5] \times [-2i, 2i] \subset \mathbb{C}$ and each points correspond to a quantum circuit M_c. Each pixel is coloured according to the probability of measuring $|1\rangle$ after applying the post-selective circuit M_c iterated n times on the initial $|0\rangle$ state. (The closer is a pixel's colour to white the higher is the probability of measuring 1.)

The Local Hamiltonian Problem

We call a Hamiltonian k-local if it can be written as a sum of local hermitian terms each of which acts non-trivially on at most k qubits. In quantum computation and information theory one usually defines the local Hamiltonians in a reduced way, such that all the local terms are orthogonal projectors. Then frustration freeness just means that there is a state lying in the kernel of all the projectors. In general deciding whether a local Hamiltonian is frustration free or not is QMA hard. (QMA is the quantum analogue of the classical complexity class NP.) To see that the problem is at least as hard as classical SAT observe that this is actually a generalisation. Consider a SAT instance on bits $b_1, \ldots, b_n$ with clauses $c_w, \ldots, c_m$. Then for a clause consisting bits $b_{j_1}, \ldots, b_{j_k}$ we define a rank-1 projector acting on qubits $q_{j_1}, \ldots, q_{j_k}$ and projecting to the one (qu)bit configuration banned by the classical clause. It is easy to see that the sum of these projector has a non-trivial kernel iff the SAT instance is satisfiable.

Although deciding frustration freeness is hard in general there exists cases in which the existence of a frustration free ground state is guaranteed. If the local projectors have not too large relative dimensions and the projectors do not share a qubit with too many other projectors (i.e. the Lovász condition holds) then we can conclude frustration freeness [6]. Recently the Lovász condition was improved to Shearer condition [8], but the corresponding proofs are existential in nature, so even if we know the existence of a frustration free ground state we do not know how to prepare it.

The corresponding classical question was solved over the past few years, and it has been proven that classically we can even find satisfying instances efficiently up to the Shearer condition [5]. The algorithm which finds a satisfying assignment is rather trivial. It is called the Moser-Tardos resampling algorithm [5], and does the following:

- Start with a uniform random assignment of the variables and a todo list consisting all caluses
- Until the todo list is not empty pick one clause from it and check if it is satisfied
  - If it is satisfied remove from the todo list
  - If it is violated then resample all the adjacent bits uniformly and add all adjacent clauses to the todo list

The quantum Moser-Tardos algorithm does the same just with bits replaced by qubits and checking replaced by projective measurement. This quantum algorithm was proven to terminate quickly in expectation under a stronger version of the Lovász condition [7]. But termination only implies successful ground state preparation if the local projectors are commuting. The natural question is whether in the non-commuting case can we also get some good convergence rate.

Simulation of the Quantum Moser-Tardos algorithm using LIQ$Ui|\rangle$

```
// Sample adjaceny list for the triangle graph
let triangleAdj = [[1;2];[1;3];[2;3]]
// (Quantum) Moser-Tardos algorithm with classical run option: decohere
let quantumMoser (ket:Ket) (adjacency:int list list) (gates:(Qubit list -> unit) list) decohere =
    let adjSet = List.map (fun myList -> Set.ofList myList) adjacency
    // Calulate which projectors share some qubits
    let gammaPlus = [for i in 0 .. adjacency.Length - 1 do
                        yield [for j in 0 .. adjacency.Length - 1 do
                                    if Set.intersect adjSet.[i] adjSet.[j]
                                        |> Set.isEmpty |> not then
                                        yield j]]
    // Initialisation - random inital values
    let reSample = reSampleFun (getRndBit [|for i in 0 .. 7 do yield 0uy|] [|8|]) ket.Rnd
    let qs = ket.Qubits
    let adjQubits = [for i in 0 .. adjacency.Length - 1 do yield getSubList (0::adjacency.[i]) qs]
    // Subroutine to check whether a projector is violated or not
    let violatedProjector id =
        // Disentangle for fast classical simulation
        if decohere then M >< qs; qs |> List.iter (fun qb -> qb.ReAnimate qb.Bit)
        gates.[id] <| adjQubits.[id]; M[qs.[0]]
        let violated = (qs.[0].Bit.v = 1)
        qs.[0].ReAnimate(CVec([|1.;0.|]))
        if violated then
            reSample <| getSubList adjacency.[id] qs
            true
        else
            false
    // The expectation value of the local Hamiltonian is simply the sum of measurement probabilities
    let expectationValues () = [ for i in 0 .. gates.Length - 1 do
                                    gates.[i] <| adjQubits.[i]
                                    yield qs.[0].Prob1
                                    gates.[i] <| adjQubits.[i]
                                    if decohere then
                                        M >< qs
                                        qs |> List.iter (fun qb -> qb.ReAnimate qb.Bit) ]
    // Recursive subroutine to check and fix all projectors
    let rec fix (remainder:int list) measured resampled =
        if remainder = [] then
            (measured,resampled,expectationValues ())
        else
            if violatedProjector remainder.[0] then
                fix ((gammaPlus.[remainder.[0]])@(remainder.Tail)) (measured+1) (resampled+1)
            else
                fix (remainder.Tail) (measured+1) resampled
    // Start running the actual procedure
    fix [0 .. gates.Length - 1] 0 0
```

The above code is an implementation of the Quantum Moser-Tardos algorithm and it has 4 arguments:
•ket: A Ket vector of qubits initialised to a random starting state.
•adjacency: A list of adjacency lists containing qubit indices on which the local projectors act.
•gates: A list of projectors provided in the form of projector controlled not gates.
•decohere: A flag indicating projectors coming from classical clauses.

Note that measuring a projector is implemented using one additional qubit namely the 0-th with the use of projector controlled not gates acting on it. So during a quantum run one only measures this qubit.

Note that if the projectors come from classical clauses then there will be no entanglement built up during iterations so it is safe to decohere the algorithm after each step in order to improve performance. This implementation has the advantage that one can run classical tests using it without too high performance trade-off if the decohere switch is used.

The code is largely self contained but contains two calls to external helper functions, one which initialises a list of qubits using the provided random number generator and another one which takes a sublist of a list for the provided indices.

Test results

I performed test runs on local projector configurations coming from various (hyper) graphs. Qubits always corresponded to the vertices and local projectors to some (hyper) edges. In the case of the cyclic graph C_n I generated random rank-1 projectors corresponding to each edges. In the case of octa- and dodecahedron I generated random projectors for all two dimensional faces with dimension 1 and 4 correspondingly. I also tried some less symmetric 3-hypergraphs with rank-1 projectors that I found using some random search. The test results can be found in Table I.

				Classical		Quantum										
				0	1	0	1	2	3	4	5	6	7	8	9	10
			M	0	7.6	0	7.6	4.3	3.8	3.6	3.6	3.3	3.2	3.2	3.1	3.1
C_3	UpB:	3	R	0	1.5	0	1.5	0.4	0.3	0.2	0.2	0.1	0.1	0.1	0.0	0.0
	Sh:	0.25	E	0.75	0	0.7577	0.1219	0.0673	0.0440	0.0312	0.0253	0.0186	0.0154	0.0119	0.0102	0.0080
			M	0	10.2	0	10.2	5.9	5.4	5.0	4.8	4.7	4.7	4.6	4.4	4.4
C_4	UpB:	6	R	0	2.1	0	2.1	0.6	0.5	0.3	0.3	0.2	0.2	0.2	0.1	0.1
	Sh:	0.125	E	0.96	0	1.0005	0.1753	0.0957	0.0693	0.0511	0.0448	0.0356	0.0303	0.0289	0.0249	0.0222
			M	0	23.7	0	22.4	14.3	12.7	11.7	11.2	10.9	11.1	10.8	10.4	10.7
C_9	UpB:	36	R	0	4.9	0	4.5	1.8	1.2	0.9	0.7	0.6	0.7	0.6	0.5	0.6
	Sh:	0.0039	E	2.20	0	2.2258	0.4448	0.2433	0.1833	0.1427	0.1229	0.1083	0.1068	0.0957	0.0872	0.0908
			M	0	27.0	0	24.7	15.9	14.0	12.9	12.7	12.1	12.0	11.4	11.7	11.4
C_{10}	UpB:	45	R	0	5.7	0	4.9	2.0	1.3	1.0	0.9	0.7	0.7	0.5	0.6	0.5
	Sh:	0.0020	E	2.52	0	2.4991	0.5052	0.2713	0.1934	0.1586	0.1418	0.1223	0.1111	0.1049	0.0976	0.0853
			M	0	20,2	0	21.4	13.6	12.1	10.8	10.0	9.7	9.7	9.8	9.4	9.1
Octahedron	UpB:	14	R	0	1.7	0	1.9	0.8	0.6	0.4	0.3	0.2	0.2	0.3	0.2	0.2
	Sh:	0.0625	E	1.01	0	1.0212	0.2204	0.1129	0.0853	0.0660	0.0563	0.0507	0.0438	0.0411	0.0340	0.0326
			M	0	23.2	0	25.8	16.1	13.2	12.5	11.9	11.7	11.4	11.4	11.0	10.9
RndTri_{10}	UpB:	389	R	0	2.0	0	2.3	0.9	0.5	0.4	0.3	0.2	0.2	0.2	0.1	0.1
	Sh:	0.0020	E	1.28	0	1.2344	0.2331	0.1171	0.0819	0.0635	0.0538	0.0444	0.0379	0.0339	0.0311	0.0278
			M	0	26.8	0	28.6	17.3	15.5	14.6	14.2	14.0	13.8	13.8	13.4	13.2
RndTri_{12}	UpB:	2236	R	0	2.4	0	2.6	0.8	0.5	0.4	0.3	0.3	0.3	0.3	0.2	0.2
	Sh:	0.0002	E	1.57	0	1.5120	0.2898	0.1410	0.0978	0.0786	0.0689	0.0563	0.0486	0.0459	0.0405	0.0374
			M	0	25.5	0	28.0	17.8	15.5	14.0	13.5	15.1	14.0	14.2	14.6	14.7
Dodecahedron	UpB:	21	R	0	2.3	0	2.7	1.0	0.6	0.3	0.3	0.5	0.3	0.4	0.4	0.5
(just 100 trials)	Sh:	0.0234	E	1.48	4e-4	1.5067	0.3640	0.1823	0.1165	0.0949	0.0752	0.0810	0.0771	0.0714	0.0741	0.0797

Table I: **Test runs of the quantum Moser-Tardos algorithm averaged over 1000 trials.**
Each row corresponds to some fixed local projector configuration and columns correspond to statistics gathered during 1000 independent trials. Each trial used a new set of random projectors (clauses) and independent random initial states. In the classical case one complete run of the Moser-Tardos algorithm already fixes everything, but in the quantum case I tested repeated applications of the Moser-Tardos algorithm to see convergence rate. The rows denoted by M, R contain the average number of projective measurements and resamplings done during the k-th run. E denotes the average of the calculated expectation value of the Hamiltonian $\langle\psi| H |\psi\rangle$ after each run. (The 0-th run just denote the random initialisation and its expectation value.) The value UpB is the classical upper bound on the number of expected resamplings given by [5], while Sh denotes the minimal guaranteed relative dimension of the kernel of the Hamiltonian i.e. the probability of a random initial state to be satisfying granted by the Shearer bound [8].

What is not shown in this table that I also did test runs where instead of using a fixed order the of projectors to test, always a random projector is chosen. The results were similar but usually worse then what this fixed order

algorithm provided so it would not add much extra information.

Note that the configurations shown here were carefully chosen to have only a little probability of initial success to make the task non-trivial. For larger system sizes it would be a less challenging requirement, but for small instances it rules out many configurations. This also means that I used instances beyond the symmetric Lovász bound, but still the number of average resamplings was very similar to the classical case, suggesting that the quantum Moser-Tardos algorithm may terminate quickly up to the Shearer bound.

Also the results in Table I suggest that the Moser-Tardos algorithm typically produces a state with relatively low expectation value after termination in the non-commutative case. However this expectation value decays slowly after repeated runs. After I spent some time thinking why can this be the case I realised the following: if there is a state which has low energy then it means that it has low overlap with the projectors, hence it is dissipated at a low rate.

After the above simulations it was still unclear what happens with the true overlap with the ground state subspace. To see the answer to this question I needed to construct a special kernel projection gate, which projects to the subspace of ground states. But this is computationally expensive, so I could not run too much tests with this setting. However the log of an interesting test trial below shows that already in the case of C_{10} one can get very poor convergence results. The log uses the same notation as Table I but has a new value P included showing the probability of kernel projection.

```
0:0000.0/Shearer guarantee on the realtive dimension of satisfying subspace : 0.001953
0:0000.0/Classical upper bound on the expected number of resamplings : 45.0
0:0000.0/Run classical test on random clauses and random input
0:0001.0/Average resamplings in 1000 simulation runs:
 0: M:  0   R: 0   E: 2.4690                1: M: 26.0 R: 5.3 E: 0.0000
0:0001.0/Run quantum test on random (real) projectors and random input
0:0001.4/Average resamplings in 100 simulation runs:
 0: M:  0   R: 0   E: 2.5177                1: M: 25.5 R: 5.2 E: 0.5132
0:0001.4/Run quantum test on random (complex) projectors and random input
0:0001.9/Average resamplings in 100 simulation runs:
 0: M:  0   R: 0   E: 2.4243                1: M: 24.7 R: 4.9 E: 0.5605
0:0001.9/Run classical test on a fixed random clause and random input
0:0002.9/Average resamplings in 1000 simulation runs:
 0: M:  0   R: 0   E: 2.5660                1: M: 32.1 R: 7.4 E: 0.0000
0:0003.0/Projectors constructed
0:0003.3/Singular values found: 1022, smallest: 0.039998, remaining relative dimension: 0.0019531
0:0003.3/Hamiltoninan constructed
0:0003.7/Kernel Gate constructed
0:0003.7/Run quantum test on a fixed (real) projector set and random input
0:0017.2/Average resamplings in 100 simulation runs:
 0: M:  0   R: 0   E: 2.6074 P: 0.0010      1: M: 22.1 R: 4.0 E: 0.4994 P: 0.0204
 2: M: 14.4 R: 1.5 E: 0.1820 P: 0.0364      3: M: 12.2 R: 0.7 E: 0.1082 P: 0.0413
 4: M: 12.3 R: 0.8 E: 0.1177 P: 0.0516      5: M: 11.3 R: 0.4 E: 0.0774 P: 0.0514
 6: M: 11.0 R: 0.3 E: 0.0642 P: 0.0588      7: M: 11.3 R: 0.4 E: 0.0615 P: 0.0643
 8: M: 10.5 R: 0.2 E: 0.0654 P: 0.0658      9: M: 10.9 R: 0.3 E: 0.0522 P: 0.0682
10: M: 10.6 R: 0.2 E: 0.0406 P: 0.0701     11: M: 10.5 R: 0.2 E: 0.0419 P: 0.0728
12: M: 10.4 R: 0.1 E: 0.0455 P: 0.0724     13: M: 10.8 R: 0.3 E: 0.0461 P: 0.0721
14: M: 10.9 R: 0.3 E: 0.0400 P: 0.0679     15: M: 10.7 R: 0.2 E: 0.0370 P: 0.0740
16: M: 10.8 R: 0.3 E: 0.0432 P: 0.0733     17: M: 10.5 R: 0.2 E: 0.0330 P: 0.0738
18: M: 10.7 R: 0.2 E: 0.0381 P: 0.0717     19: M: 10.6 R: 0.2 E: 0.0374 P: 0.0727
20: M: 10.6 R: 0.2 E: 0.0264 P: 0.0716
```

Conclusions

I could gain significant insight into the performance of the quantum Moser-Tardos algorithm in the general non-commuting case. I hope that later I will even be able to prove some analytical results based on the hints I got from the LIQ$Ui|\rangle$ simulation. My main conclusion are the following:

•The quantum Moser-Tardos algorithm seems to terminate efficiently up to the Shearer bound

•The quantum Moser-Tardos algorithm seems to be efficient for producing states with low expectation value

•The eigenvalue gap should play a key role in the analysis of the quantum Moser-Tardos algorithm

+1 Suggestion: The performance of several LIQ$Ui|\rangle$ simulations could be improved by enabling the use of matrices which consists of both sparse and low rank parts. For example it would be the case for my kernel projection gate. Note that projectors are also widely used in other quantum algorithms e.g. in quantum walk based search algorithms.

[1] Gilyén, A., Kiss, T. & Jex, I. Exponential Sensitivity and its Cost in Quantum Physics. *Scientific Reports* **6**, 20076 (2016)

[2] Lloyd, S. & Slotine, J. E. Quantum feedback with weak measurements. *Phys. Rev. A* **62,** 012307 (2000)

[3] Verstraete, F., Wolf, M. M., & Cirac, J. I. Quantum computation and quantum-state engineering driven by dissipation. *Nature Physics* **5**, 633-636 ((2009)

[4] Szegedy, M. The Lovász Local Lemma – A Survey. In *Computer Science – Theory & Appl.*, pp. 1-11., Springer (2013)

[5] Kolipaka, K. B. R., Szegedy, M. Moser and tardos meet Lovász. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC '11)*, pp. 235–244., ACM (2011)

[6] Ambainis, A., Kempe, J., Sattath, O. On the restrictiveness of quantum satisfiability. *JACM* **59**, i. 5, a.n. 24 (2012)

[7] Schwarz, M., Cubitt, T. S., Verstraete, F. An Information-Theoretic Proof of the Constructive Commutative Quantum Lovász Local Lemma. *arxiv:1311.6474*, (2013)

[8] Sattath, O. A quantum Lovász local lemma. *Unpublished manuscript* (2015)

[9] Harrow, A. W., Hassidim, A., Lloyd, S. Quantum algorithm for solving linear systems of equations, Phys. Rev. Lett. 103, 150502 (2009)