Title: Dissipative Quantum Gibbs Sampling

Speakers: Daniel Zhang

Series: Perimeter Institute Quantum Discussions

Date: May 31, 2023 - 11:00 AM

URL: https://pirsa.org/23050158

Abstract:

Systems in thermal equilibrium at non-zero temperature are described by their Gibbs state. For classical many-body systems, the Metropolis-Hastings algorithm gives a Markov process with a local update rule that samples from the Gibbs distribution. For quantum systems, sampling from the Gibbs state is significantly more challenging. Many algorithms have been proposed, but these are more complex than the simple local update rule of classical Metropolis sampling, requiring non-trivial quantum algorithms such as phase estimation as a subroutine.

Here, we show that a dissipative quantum algorithm with a simple, local update rule is able to sample from the quantum Gibbs state. In contrast to the classical case, the quantum Gibbs state is not generated by converging to the fixed point of a Markov process, but by the states generated at the stopping time of a conditionally stopped process. This gives a new answer to the long-sought-after quantum analogue of Metropolis sampling. Compared to previous quantum Gibbs sampling algorithms, the local update rule of the process has a simple implementation, which may make it more amenable to near-term implementation on suitable quantum hardware. We also show how this can be used to estimate partition functions using the stopping statistics of an ensemble of runs of the dissipative Gibbs sampler. This dissipative Gibbs sampler works for arbitrary quantum Hamiltonians, without any assumptions on or knowledge of its properties, and comes with certifiable precision and run-time bounds.
This talk is based on 2304.04526, completed in collaboration with Jan-Lukas Bosse and Toby Cubitt.

Zoom Link: https://pitp.zoom.us/j/96780945341?pwd=NG9SUjE4SkVia3VqazNXUFNUamhRdz09

Outline:

1. Background & Introduction
2. The Algorithm
3. Bonus: Partition Function Estimation

PHASECRAFT

The Classical Case:

For a classical statistical mechanical many body system, Hamiltonian $H(x)$ is a scalar function of the many-body state $x = (x_i)$ (*e.g.* a list of spins).

The Boltzmann distribution:

$$\Pr(x) = \frac{e^{-\beta H(x)}}{\mathcal{Z}} \qquad \mathcal{Z} = \sum_x e^{-\beta H(x)}$$

This is already #P-hard. Quantum case, expected to be even harder [Bravyi, Chowdhury, Gosset, Wocjan '21]

Uses: Gibbs distributions have applications in machine learning (Boltzmann machines), Markov chain Monte Carlo methods, optimisation…

PHASECRAFT

Metropolis-Hastings: [Metropolis '53; Hastings '70]

1. Starting from arbitrary x, propose x' from symmetric proposal distribution $q(x' \mid x)$, *e.g.* flip randomly chosen x_i.
2. Compute $p = e^{-\beta H(x')} / e^{-\beta H(x)}$.
3. Accept x' with probability $\min(p, 1)$.

- MH generates a Markov chain whose stationary distribution is the Gibbs distribution. That is, once the Markov chain has converged, the states it generates will be distributed according to Gibbs.
- Note, if $H(x)$ is local, then computing $p = e^{-\beta H(x')} / e^{-\beta H(x)}$ is local. Thus the update rule and transition probabilities of MH are ***local***.

PHASECRAFT

The Quantum Case:

MH required evaluation of $H(x)$, classically this is just an easy function evaluation. The local update rule is also easy to implement.

Trying to generalise to the quantum case presents two immediate difficulties:

- One needs to evaluate $\langle\psi|H|\psi\rangle$, given a single copy of $|\psi\rangle$. This requires an expensive algorithm such as quantum phase estimation.
- One needs to decide whether to accept a proposed state $|\psi'\rangle$ or revert to the original $|\psi\rangle$. However, QPE projects onto an energy eigenstate. It's not obvious how to do this!

Original quantum metropolis algorithm [Temme, Osborne, Vollbrecht, Poulin, Verstraete '11] solves second issue (with Marriott-Watrous rewinding). Still, requires running complex quantum circuit (for QPE).

PHASECRAFT

Many subsequent works (too many to list!), falling into a few categories:

Method:	Requirements:
Grover/related modern quantum algorithm techniques applied to quantum metropolis sampling (polynomial speedups)	Large, global quantum circuits.
Imaginary time evolution methods	Large, global quantum circuits.
Variational approaches (QAOA, quantum Boltzmann machines, variational ITE…)	Heuristic (requires ansatz), non-trivial classical optimisation over quantum circuits.
Emulated thermalisation via coupling to thermal bath of ancilla	Large supply of fresh ancilla, simulation of time evolution of large system-bath Hamiltonian.
Emulated thermalisation via Davies generator (or similar Lindbladian), quantum Markov semigroup whose fixed point is the Gibbs state: $e^{\mathscr{L}^\dagger t}\rho_G = \rho_G$	Large, global quantum circuits (QPE, Operator Fourier Transform…)

Approach taken by [Chen, Kastoryano, Brandão, Gilyén '23], current best dependence of runtime on precision (which we also achieve). See also there for comprehensive review of existing methods!

PHASECRAFT

Method:

Grover/related modern quantum algorithm techniques applied to quantum metropolis sampling (polynomial speedups)

Imaginary time evolution methods

Variational approaches (QAOA, quantum Boltzmann machines, variational ITE…)

Emulated thermalisation via coupling to thermal bath of ancilla

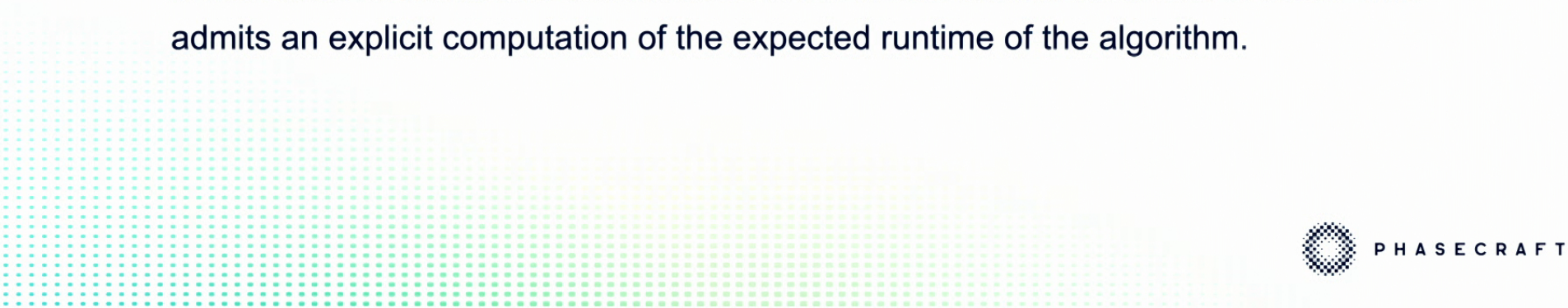
Emulated thermalisation via Davies generator (or similar Lindbladian), quantum Markov semigroup whose fixed point is the Gibbs state: $e^{\mathscr{L}^\dagger t}\rho_G = \rho_G$

Many of these existing methods have a runtime which depends on the mixing time of a Markov chain, which is general difficult to determine (no closed form expressions), apart from in some specialised systems (*e.g.* commuting case)

PHASECRAFT

Our approach for Gibbs sampling comes from the same family of algorithms introduced recently in [Cubitt '23] for ground state sampling - the **Dissipative Quantum Eigensolver**. It inherits some desirable features:

- Each step in the algorithm consists of performing weak measurements of ***local*** terms in the Hamiltonian, and then resampling based on the outcome.
- Instead of running a quantum Markov chain until it converges to its fixed point, a probabilistic stopping rule conditioned on the measurement outcomes is used. This admits an explicit computation of the expected runtime of the algorithm.



PHASECRAFT

Quantum Instrument:

- Formulation of quantum measurement.
- In the two outcomes case $x \in \{0,1\}$, measurement via a quantum instrument $\mathcal{I}$ is specified by two CP maps $\{\mathcal{E}_0, \mathcal{E}_1\}$ such that the sum $\mathcal{E}_0 + \mathcal{E}_1$ is trace preserving and:

Outcome probability: $p(x|\rho) = \mathrm{Tr}\mathcal{E}_x(\rho)$ **Post-measurement state:** $\frac{\mathcal{E}_x(\rho)}{\mathrm{Tr}\mathcal{E}_x(\rho)}$

- Can be described as a quantum channel: $\rho \mapsto \mathcal{I}(\rho) = \sum_x \mathcal{E}_x(\rho) \otimes \underbrace{|x\rangle\langle x|}_{\text{classical register}}$

followed by projective measurement on classical register.

PHASECRAFT

Dissipative Gibbs Sampler

First, the most general form of the algorithm:

Algorithm (Dissipative Gibbs Sampler): for a local Hamiltonian $H = \sum_{i=1}^{m} h_i$, define the quantum instrument:

$$\mathscr{E}_0(\rho) = K\rho K, \qquad \mathscr{E}_1(\rho) = (1 - \mathrm{Tr}(K\rho K))\rho_0$$

where K is a Hermitian operator ($K^2 < \mathbb{I}$) and $\|K - f(H)\| \leq \epsilon$ for some injective function f.

Let $0 \leq r_n \leq 1$ be a set of probabilities, $n = 0,1,2,\ldots$.

The DGS algorithm consists of successively applying instrument $\{\mathscr{E}_0, \mathscr{E}_1\}$, starting from an initial state ρ_0 and after a run of n zeros, stopping with probability r_n or continuing running with probability $1 - r_n$.

We will show that by choosing the biased coins $\{r_n\}$ appropriately, and the operator K, we can sample from the Gibbs state with the promised properties (precision, local implementation *etc*).

PHASECRAFT

Dissipative Gibbs Sampler

$$\mathscr{E}_0(\rho) = K\rho K, \qquad \underbrace{\mathscr{E}_1(\rho) = (1 - \mathrm{Tr}(K\rho K))\rho_0}$$

If the "wrong" outcome 1 is obtained, resample the entire state to the initial state ρ_0.

In practice we will take this to be the maximally mixed state $\rho_0 = \frac{I}{D}$.

Our strategy will be first to derive expected output states and runtimes of the DGS algorithm for general $K, \{r_n\}$, before specialising to the ones of interest.

PHASECRAFT

Expected Output State:

What is the expected output state $\mathbb{E}[\rho_\tau]$ at the stopping time τ of the DGS algorithm?

Warm-up: a fair coin is tossed repeatedly until five heads occurs? What is the expected number E of tosses?

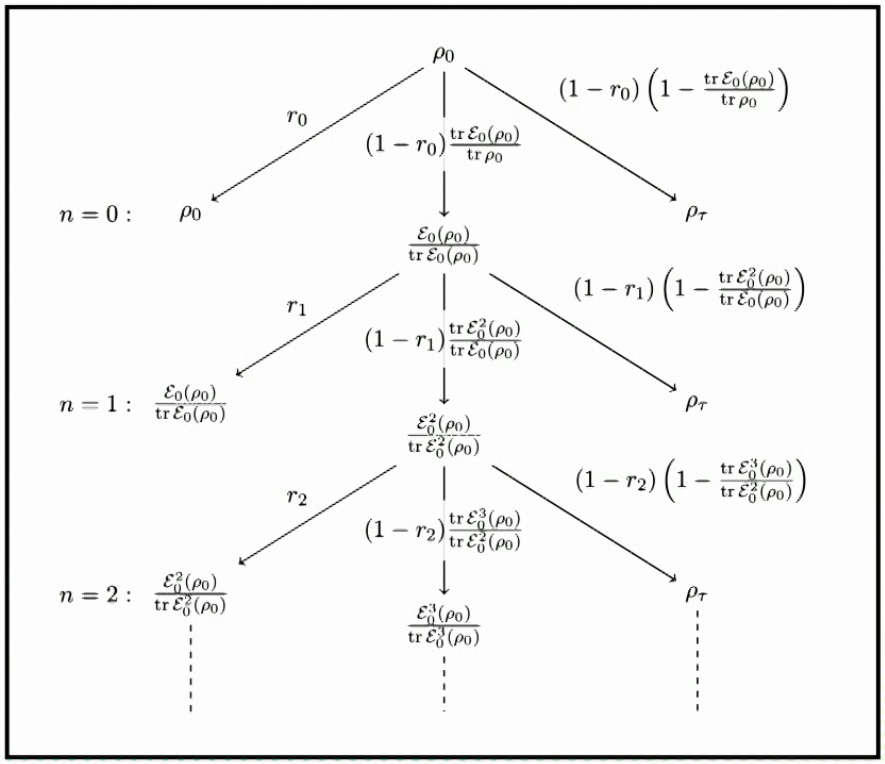
$$E = \frac{1}{2}(E+1) + \frac{1}{4}(E+2) + \frac{1}{8}(E+3) + \frac{1}{16}(E+4) + \frac{1}{32}(E+5) + \frac{1}{32}(5)$$

$T\ldots$ $HT\ldots$ $HHT\ldots$ $HHHT\ldots$ $HHHHT\ldots$ $HHHHH$

$$\Rightarrow E = 62.$$

PHASECRAFT

Expected Output State:



$$
\begin{aligned}
\mathbb{E}[\rho_\tau] &= r_0\rho_0 + (1-r_0)\left(1-\frac{\mathrm{tr}\,\mathcal{E}_0(\rho_0)}{\mathrm{tr}\,\rho_0}\right)\mathbb{E}[\rho_\tau] \\
&\quad + (1-r_0)r_1\frac{\mathcal{E}_0(\rho_0)}{\mathrm{tr}\,\rho_0} + (1-r_0)(1-r_1)\left(\frac{\mathrm{tr}\,\mathcal{E}_0(\rho_0)}{\mathrm{tr}\,\rho_0} - \frac{\mathrm{tr}\,\mathcal{E}_0^2(\rho_0)}{\mathrm{tr}\,\rho_0}\right)\mathbb{E}[\rho_\tau] \\
&\quad + (1-r_0)(1-r_1)r_2\frac{\mathcal{E}_0^2(\rho_0)}{\mathrm{tr}\,\rho_0} + (1-r_0)(1-r_1)(1-r_2)\left(\frac{\mathrm{tr}\,\mathcal{E}_0^2(\rho_0)}{\mathrm{tr}\,\rho_0} - \frac{\mathrm{tr}\,\mathcal{E}_0^3(\rho_0)}{\mathrm{tr}\,\rho_0}\right)\mathbb{E}[\rho_\tau] \\
&\quad + \dots \\
&= \sum_{n=0}^{\infty} r_n R_n \frac{\mathcal{E}_0^n(\rho_0)}{\mathrm{tr}\,\rho_0} + \mathbb{E}[\rho_\tau]\sum_{n=0}^{\infty} R_{n+1}\left(\frac{\mathrm{tr}\,\mathcal{E}_0^n(\rho_0)}{\mathrm{tr}\,\rho_0} - \frac{\mathrm{tr}\,\mathcal{E}_0^{n+1}(\rho_0)}{\mathrm{tr}\,\rho_0}\right),
\end{aligned}
$$

$$
R_n = \prod_{j=0}^{n-1}(1-r_j)
$$

PHASECRAFT

Expected Output State:

What is the expected output state $\mathbb{E}[\rho_\tau]$ at the stopping time τ of the DGS algorithm?

Warm-up: a fair coin is tossed repeatedly until five heads occurs? What is the expected number E of tosses?

$$E = \frac{1}{2}(E+1) + \frac{1}{4}(E+2) + \frac{1}{8}(E+3) + \frac{1}{16}(E+4) + \frac{1}{32}(E+5) + \frac{1}{32}(5)$$

$T\ldots$ $HT\ldots$ $HHT\ldots$ $HHHT\ldots$ $HHHHT\ldots$ $HHHHH$

$$\Rightarrow E = 62.$$

PHASECRAFT

Expected Output State:

Lemma: The expected output state of the DGS algorithm is given by:

$$\mathbb{E}[\rho_\tau] = \frac{\sum_{n=0}^{\infty} r_n R_n \mathscr{E}_0^n(\rho_0)}{\sum_{n=0}^{\infty} r_n R_n \mathrm{Tr}\,\mathscr{E}_0^n(\rho_0)}$$

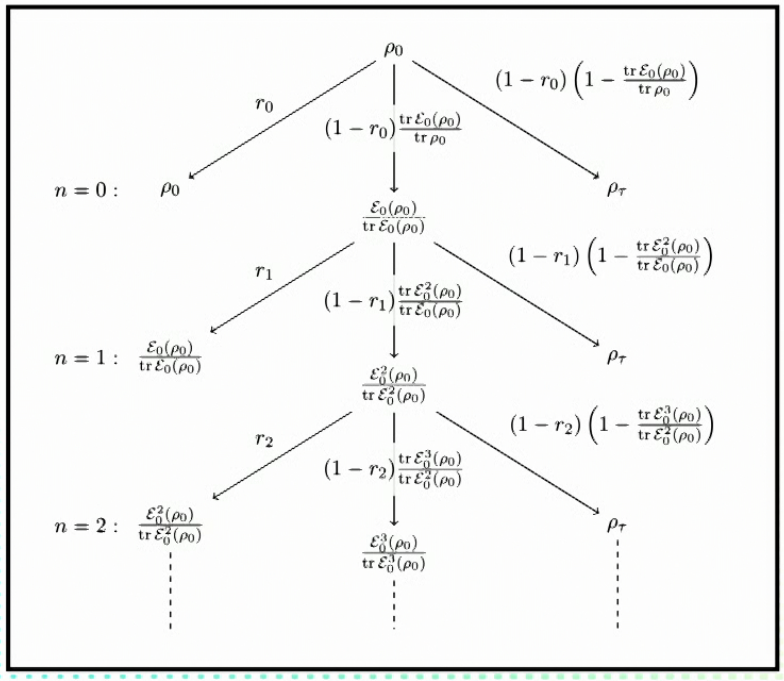
where $R_n = \prod_{j=0}^{n-1}(1 - r_j)$.

N.B. The normalisations by $\mathrm{Tr}(\mathscr{E}_0^n(\rho_0))$ drop out in the expected state! It's feasible then if we tune the r_n (and hence R_n) appropriately, and K in $\mathscr{E}_0$ to some appropriate function of H, that we will generate an expected state close to the Gibbs state.

PHASECRAFT

Expected Runtime:

Before moving on, note that the expected runtime $\mathbb{E}[\tau]$ of the DGS algorithm can be computed in exactly the same way.



Replace each leaf on the n^{th} level on the LEFT by $n+1$ (the number of steps required to reach there).

Replace each leaf on the n^{th} level on the RIGHT by $\mathbb{E}[\tau]+n+1$.

Now, sum over all the leaves, weighted by the probability of reaching them:

$$\mathbb{E}[\tau] = \sum_{n=0}(n+1)r_n R_n \frac{\operatorname{tr}\mathcal{E}_0^n(\rho_0)}{\operatorname{tr}\rho_0} + (\mathbb{E}[\tau]+n+1)R_{n+1}\left(\frac{\operatorname{tr}\mathcal{E}_0^n(\rho_0)}{\operatorname{tr}\rho_0} - \frac{\operatorname{tr}\mathcal{E}_0^{n+1}(\rho_0)}{\operatorname{tr}\rho_0}\right)$$

$$\Rightarrow \quad \mathbb{E}[\tau] = \frac{\sum_{n=0}^{\infty} R_n \mathrm{Tr}\ \mathscr{E}_0^n(\rho_0)}{\sum_{n=0}^{\infty} r_n R_n \mathrm{Tr}\mathscr{E}_0^n(\rho_0)}$$

PHASECRAFT

Towards the Gibbs state...

Recall $R_n = \prod_{i=1}^{n-1}(1 - r_i)$ and $\mathscr{E}_0(\rho) = K\rho K$

$$\mathbb{E}[\rho_\tau] = \frac{\sum_{n=0}^{\infty} r_n R_n \mathscr{E}_0^n(\rho_0)}{\sum_{n=0}^{\infty} r_n R_n \mathrm{Tr}\,\mathscr{E}_0^n(\rho_0)} \xrightarrow{\rho_0 = \mathbb{I}/D} \mathbb{E}[\rho_\tau] = \frac{\sum_{n=0}^{\infty} r_n R_n K^{2n}}{\sum_{n=0}^{\infty} r_n R_n \mathrm{Tr}K^{2n}} \xrightarrow{r_n R_n \propto \frac{\lambda^{2n}}{(2n)!} \text{ for constant } \lambda} \mathbb{E}[\rho_\tau] = \frac{\cosh \lambda K}{\mathrm{Tr}\cosh \lambda K}$$

N.B. it is not hard to show that it is possible to choose coin-flip probabilities $0 \leq r_n \leq 1$ such that $r_n R_n \propto \frac{\lambda^{2n}}{(2n)!}$ is satisfied. Solving recursively:

$$r_n = \frac{r_0 \frac{\lambda^{2n}}{(2n)!}}{1 - \sum_{j=0}^{n-1} r_0 \frac{\lambda^{2j}}{(2j)!}}$$

Then $0 \leq r_n \leq 1$ if and only if $r_0 \leq \frac{1}{\cosh \lambda}$. Intuitively clear that runtime is minimised if r_n (which are monotonic in r_0) are maximised. Thus choose $r_0 = \frac{1}{\cosh \lambda}$.

PHASECRAFT

Why did we make those choices? Let $\kappa = \sum_i \|h_i\|$, an upper bound on $\|H\|$.

Then if we choose $K = (1-\epsilon)^{2m-1}\left(1 - \frac{\epsilon}{\kappa}H\right)$ and $\lambda = \frac{\beta\kappa}{\epsilon(1-\epsilon)^{2m-1}}$, the expected output state:

$$\mathbb{E}[\rho_\tau] = \frac{\cosh \lambda K}{\mathrm{Tr}\cosh \lambda K} = \frac{\rho_G}{1 + e^{-\frac{2\beta\kappa}{\epsilon}}\frac{\mathrm{Tr}e^{\beta H}}{\mathrm{Tr}e^{-\beta H}}} + \frac{e^{-\frac{2\beta\kappa}{\epsilon}}e^{\beta H}}{\mathrm{Tr}(e^{-\beta H} + e^{-\frac{2\beta\kappa}{\epsilon}}e^{\beta H})},,$$

then we have that:

$$\|E[\rho_\tau] - \rho_G\|_1 = O(e^{-\frac{\beta\kappa}{\epsilon}}).$$

The DGS algorithm produces a state close to the Gibbs state! "Cosh is close to the exponential…."

PHASECRAFT

Note:

- In $K = (1-\epsilon)^{2m-1}(1-\epsilon H/\kappa)$, the prefactor $(1-\epsilon)^{2m-1}$ is arbitrary at this stage, as long as it is < 1.
- This K is also used in the **Dissipative Quantum Eigensolver** [Cubitt '23]; in that context it is important that K is an AGSP (**Approximate Ground State Projector**), originally introduced for proving area-laws in many body systems in [Arad, Kitaev, Landau, Vazirani '13].
- For us, we're just using the fact that K is linear (injective?) in H, and that it is Hermitian and obeys $K^2 \leq \mathbb{I}$, so that it defines a good quantum instrument/measurement.
- Importantly: implementing $\mathscr{E}_0(\rho) = K\rho K$ requires ***global*** measurements. We want to upgrade this to only requiring local measurements.

PHASECRAFT

$$H = \sum_{i=1}^{m} h_i$$

Lemma [Cubitt '23]: Let:

$$K' = \prod_{i=1}^{m} ((1-\epsilon)\mathbb{I} + \epsilon\kappa_i k_i) \prod_{i=m}^{1} ((1-\epsilon)\mathbb{I} + \epsilon\kappa_i k_i)$$

where:

$$k_i = \frac{1 - h_i/\|h_i\|}{2}, \quad \kappa_i = \frac{\|h_i\|}{\kappa}, \quad \kappa = \sum_i \|h_i\|$$

then:

$$\|K' - K\| = O(\epsilon^2 m^2).$$

K' corresponds to weak-measuring ***local*** terms of the Hamiltonian in a prescribed order, and approximates the K from before!

PHASECRAFT

Running the DGS algorithm with $\mathscr{E}_0(\rho) = K'\rho K'$ will generate a state close to the Gibbs state of a perturbed Hamiltonian H':

$$\|\mathbb{E}[\rho_\tau] - \rho_G(H')\|_1 = O(e^{-\frac{\beta\kappa}{\epsilon}}),$$

where the perturbed Hamiltonian H' obeys:

$$\|H' - H\| = O(\epsilon m^2 \kappa).$$

This is because, using the previous Lemma, we may write:

$$K' = (1-\epsilon)^{2m-1}(1 - \epsilon H'/\kappa)$$

i.e. mimicking the form of $K = (1-\epsilon)^{2m-1}(1-\epsilon H/\kappa)$ from before, and apply the same arguments as before.

[Poulin & Wocjan '09] tells us that if two Hamiltonians are close, then so are their Gibbs states:

If $\|H' - H\| = O(\epsilon^2 m^2 \kappa)$, then:

$$\|\rho_G(H') - \rho_G(H)\|_1 = O(\beta \epsilon \kappa m^2).$$

Using the triangle inequality:

$$\|\mathbb{E}[\rho_\tau] - \rho_G(H)\|_1 = \underbrace{O(e^{-\frac{\beta\kappa}{\epsilon}})}_{\text{Much smaller!}} + O(\beta \epsilon \kappa m^2)$$

PHASECRAFT

So we're done!

Theorem (Dissipative Gibbs Sampler): for a local Hamiltonian $H = \sum_{i=1}^{m} h_i$, define the quantum instrument:

$$\mathscr{E}_0(\rho) = K\rho K, \qquad \mathscr{E}_1(\rho) = (1 - \mathrm{Tr}(K\rho K))\rho_0$$

Where :

$$K = \prod_{i=1}^{m}((1-\epsilon)\mathbb{I} + \epsilon\kappa_i k_i)\prod_{i=m}^{1}((1-\epsilon)\mathbb{I} + \epsilon\kappa_i k_i) \quad k_i = \frac{1 - h_i/\|h_i\|}{2}, \quad \kappa_i = \frac{\|h_i\|}{\kappa}, \quad \kappa = \sum_i \|h_i\|$$

$$\text{and } r_n = \frac{r_0 \frac{\lambda^{2n}}{(2n)!}}{1 - \sum_{j=0}^{n-1} r_0 \frac{\lambda^{2j}}{(2j)!}} .$$

Successively apply instrument $\{\mathscr{E}_0, \mathscr{E}_1\}$, starting from an initial state $\rho_0 = \mathbb{I}/D$ and after a run of n zeros, stopping with probability r_n or continuing running with probability $1 - r_n$. Then the expected state at the stopping time τ satisfies:

$$\|\mathbb{E}[\rho_\tau] - \rho_G\|_1 = O(\beta\epsilon\kappa m^2)$$

PHASECRAFT

The Runtime:

Plugging in the λ, K *etc* into the explicit expression for the DGS algorithm:

$$\mathbb{E}[\tau] = \frac{\cosh(\lambda)\mathrm{Tr}\left(\frac{1}{1-K^2}\right)}{\mathrm{Tr}(\cosh(\lambda K))} - \frac{\mathrm{Tr}\left(\frac{K^2\cosh(\lambda K)}{1-K^2}\right)}{\mathrm{Tr}(\cosh(\lambda K))} \leq \ldots. \leq \underbrace{\frac{6}{\epsilon} e^{\frac{2\beta\kappa m}{(1-\epsilon)^{2m-1}}}}_{\text{Very coarse upper bound}}.$$

- The dependence on precision ϵ is optimal amongst current methods (matches [Chen, Kastoryano, Brandão, Gilyén '23]).
- Overall runtime is hard to compare because of lack of analytic results on mixing times of quantum Markov chains.
- The proof: one ends up bounding infinite series in K^2. Recall $K = \prod_{i=1}^{m}((1-\epsilon)\mathbb{I} + \epsilon\kappa_i k_i)\prod_{i=m}^{1}((1-\epsilon)\mathbb{I} + \epsilon\kappa_i k_i)$. Use submultiplicativity of the operator norm: $\|K\| \leq \left[\prod_{i=1}^{m}(1-\epsilon) + \epsilon\frac{\|h_i\|}{\kappa}\right]^2 \leq \left[\frac{1}{m}\left(\sum_{i=1}^{m}(1-\epsilon) + \epsilon\frac{\|h_i\|}{\kappa}\right)\right]^{2m} = \left(1 - \frac{m-1}{m}\epsilon\right)^{2m}$.

PHASECRAFT

3. Bonus: Partition Function Estimation

PHASECRAFT

3. Bonus: Partition Function Estimation

PHASECRAFT

Obtaining The Partition Function

- For the purposes of simulation, we want to estimate expectation values: $\mathrm{Tr}\rho\mathcal{O}$ of observables.
- To do so, need to produce multiple samples of the Gibbs state, running the DGS algorithm multiple times.
- If you keep track of the stopping statistics of the algorithm for the multiple runs, you can obtain an estimate for the partition function $\mathcal{Z} = \mathrm{Tr}e^{-\beta H}$ for free!

$$\left| De^{\beta\kappa(2m-1)}\mathbb{E}\left[\frac{\#\text{ runs}}{\#\text{ state resets}}\right] - \mathcal{Z}(H)\right| = \underbrace{O(\mathcal{Z}(H)\epsilon\kappa m^2\beta)}_{\text{multiplicative error}},$$

- "Run" - running DGS algorithm until the stopping rule tells us to stop.
- "State Resets" - the number of times the measurement outcome is 1 and we are told to reset to the maximally mixed state, or we restart the algorithm after obtaining a sample.

PHASECRAFT

The Runtime:

Plugging in the λ, K *etc* into the explicit expression for the DGS algorithm:

$$\mathbb{E}[\tau] = \frac{\cosh(\lambda)\mathrm{Tr}\left(\frac{1}{1-K^2}\right)}{\mathrm{Tr}(\cosh(\lambda K))} - \frac{\mathrm{Tr}\left(\frac{K^2\cosh(\lambda K)}{1-K^2}\right)}{\mathrm{Tr}(\cosh(\lambda K))} \leq \ldots. \leq \underbrace{\frac{6}{\epsilon} e^{\frac{2\beta\kappa m}{(1-\epsilon)^{2m-1}}}}_{\text{Very coarse upper bound}}.$$

- The dependence on precision ϵ is optimal amongst current methods (matches [Chen, Kastoryano, Brandão, Gilyén '23]).
- Overall runtime is hard to compare because of lack of analytic results on mixing times of quantum Markov chains.
- The proof: one ends up bounding infinite series in K^2. Recall $K = \prod_{i=1}^{m}((1-\epsilon)\mathbb{I} + \epsilon\kappa_i k_i)\prod_{i=m}^{1}((1-\epsilon)\mathbb{I} + \epsilon\kappa_i k_i)$. Use submultiplicativity of the operator norm: $\|K\| \leq \left[\prod_{i=1}^{m}(1-\epsilon) + \epsilon\frac{\|h_i\|}{\kappa}\right]^2 \leq \left[\frac{1}{m}\left(\sum_{i=1}^{m}(1-\epsilon) + \epsilon\frac{\|h_i\|}{\kappa}\right)\right]^{2m} = \left(1 - \frac{m-1}{m}\epsilon\right)^{2m}$.

PHASECRAFT

PHASECRAFT

Thanks for listening!

PHASECRAFT.IO