Title: Faster quantum and classical SDP approximations for quadratic binary optimization

Speakers: Richard Kueng

Series: Perimeter Institute Quantum Discussions

Date: October 28, 2019 - 2:00 PM

URL: http://pirsa.org/19100088

Abstract: We give a quantum speedup for solving the canonical semidefinite programming relaxation for binary quadratic optimization. The class of relaxations for combinatorial optimization has so far eluded quantum speedups. Our methods combine ideas from quantum Gibbs sampling and matrix exponent updates. A de-quantization of the algorithm also leads to a faster classical solver. For generic instances, our quantum solver gives a nearly quadratic speedup over state-of-the-art algorithms.
This is joint work with Fernando Brandao (Caltech) and Daniel Stilck Franca (QMATH, Copenhagen).

Caltech

quantum SDP speedups

Richard Küng

Faster quantum and classical SDP approximations for quadratic binary optimization

arXiv:1909.04613

F.G.S.L. Brandão[1,2], R. Kueng[1,2] and D. Stilck França[3]

[1]Department of Computing + Mathematical Sciences (CMS)
California Institute of Technology

[2]Institute for Quantum Information and Matter (IQIM)
California Institute of Technology

[3]QMATH, Department of Mathematical Sciences
University of Copenhagen

October 28, 2019

Table of Contents

Caltech

quantum SDP speedups

Richard Küng

Motivation
The problem
Meta-algorithm
i. Optimization ⇒ feasibility
ii. Gibbs substitution
iii. Hamiltonian Updates
Runtime analysis
Convergence
Classical runtime
Quantum runtime
Summary

Table of Contents

Caltech

Prospective applications of quantum computers

Caltech

Feynman: simulate microscopic systems

- quantum chemistry, field theories, . . .
- prepare Gibbs states

Prospective applications of quantum computers

Caltech

Feynman: simulate microscopic systems

- quantum chemistry, field theories, . . .
- prepare Gibbs states

Shor: solve expensive computing problems

- factoring, discrete logarithm

Prospective applications of quantum computers

Feynman: simulate microscopic systems

- quantum chemistry, field theories, . . .
- prepare Gibbs states

Shor: solve expensive computing problems

- factoring, discrete logarithm

Grover: search data bases

Prospective applications of quantum computers

Caltech

Feynman: simulate microscopic systems

- quantum chemistry, field theories, . . .
- prepare Gibbs states

Shor: solve expensive computing problems

- factoring, discrete logarithm

Grover: search data bases

Brandão & Svore: solve (certain) optimization problems quicker

- linear programming, certain SDPs
- **here:** SDP relaxations for binary quadratic optimization

Prospective applications of quantum computers

Caltech

Feynman: simulate microscopic systems
- quantum chemistry, field theories, . . .
- prepare Gibbs states

Shor: solve expensive computing problems
- factoring, discrete logarithm

Grover: search data bases

Brandão & Svore: solve (certain) optimization problems quicker
- linear programming, certain SDPs
- **here:** SDP relaxations for binary quadratic optimization
  - ⇒ CutNorm (MaxCut)
  - ⇒ community detection
  - ⇒ semi-discrete matrix factorization

Underlying idea

Embed quantum simulation as fast subroutine into powerful classical solvers.

Table of Contents

Caltech

Convex optimization problems

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m \end{aligned}$$

convex if $\mathcal{C}_i$ are convex sets and f is concave function

convex problems are (often) computationally tractable

Convex optimization problems

Caltech

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{maximize}} & f(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_m \end{array}$$

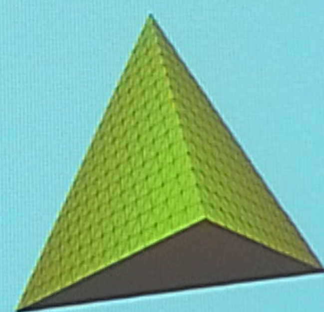
convex if $\mathcal{C}_i$ are convex sets and f is concave function

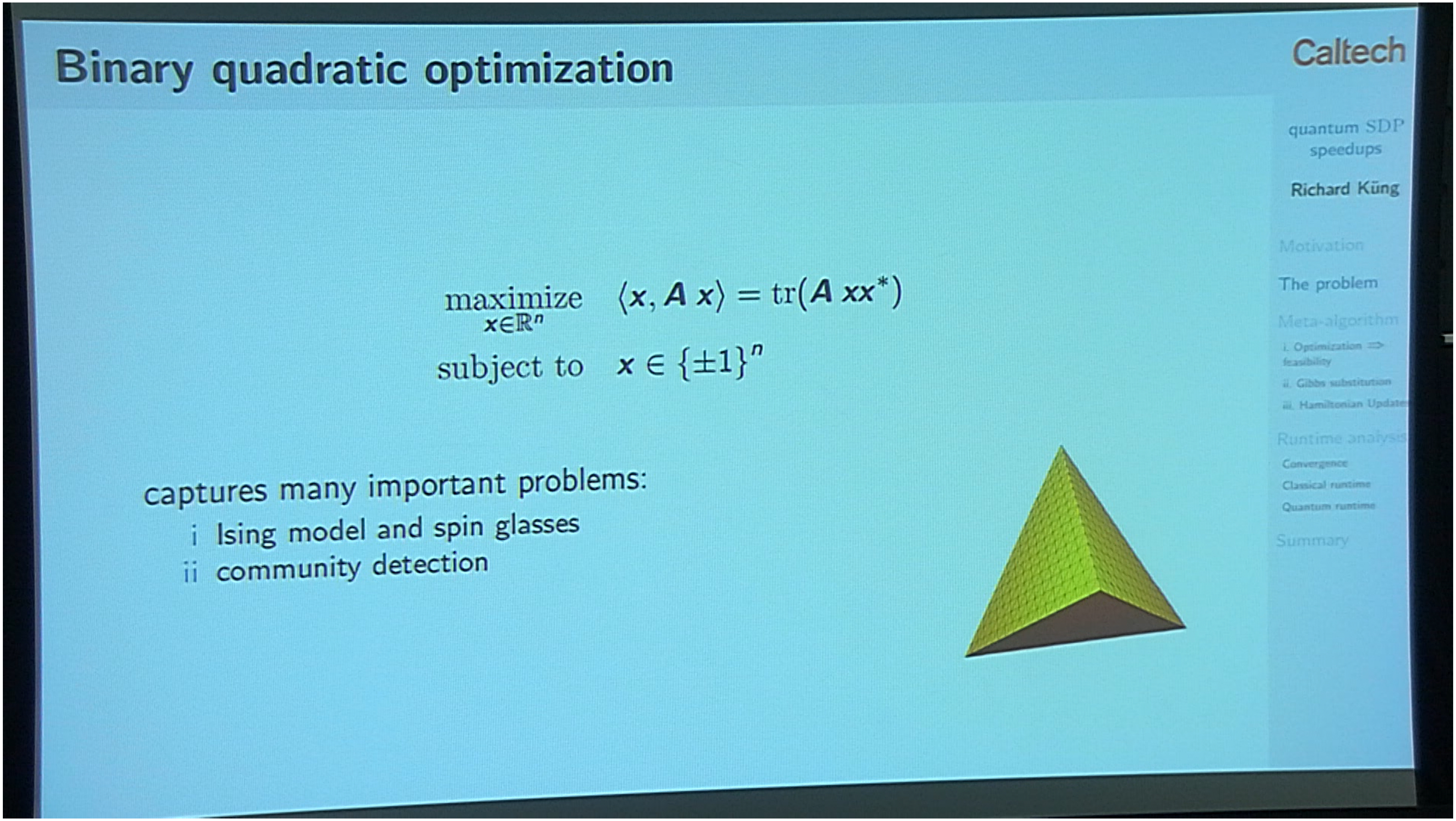
convex problems are (often) computationally tractable

general purpose solvers are slow

Binary quadratic optimization

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{maximize}} \quad & \langle x, A\, x \rangle = \mathrm{tr}(A\, x x^*) \\ \text{subject to} \quad & x \in \{\pm 1\}^n \end{aligned}$$




Binary quadratic optimization
Caltech
quantum SDP speedups
Richard Küng
Motivation
The problem
Meta-algorithm
i. Optimization ⟹ feasibility
ii. Gibbs substitution
iii. Hamiltonian Updates
Runtime analysis
Convergence
Classical runtime
Quantum runtime
Summary
$\underset{x \in \mathbb{R}^n}{\text{maximize}} \quad \langle x, A\, x \rangle = \mathrm{tr}(A\, xx^*)$
$\text{subject to} \quad x \in \{\pm 1\}^n$
captures many important problems:
i Ising model and spin glasses
ii community detection

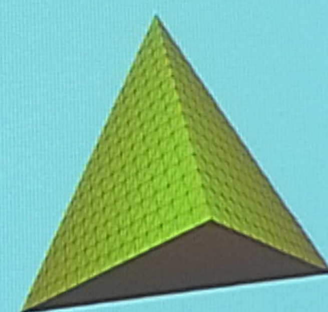
Binary quadratic optimization

Caltech

$$\underset{x \in \mathbb{R}^n}{\text{maximize}} \quad \langle x, A\, x \rangle = \mathrm{tr}(A\, x x^*)$$
$$\text{subject to} \quad x \in \{\pm 1\}^n$$

captures many important problems:

i Ising model and spin glasses
ii community detection
iii MaxCut and CutNorm



Binary quadratic optimization

Caltech

quantum SDP speedups

Richard Küng

$$\underset{x \in \mathbb{R}^n}{\text{maximize}} \quad \langle x, A\, x \rangle = \mathrm{tr}(A\, xx^*)$$

$$\text{subject to} \quad x \in \{\pm 1\}^n$$

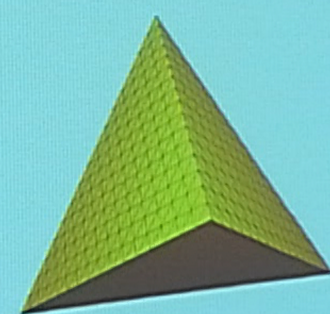
captures many important problems:

i. Ising model and spin glasses

ii. community detection

iii. MaxCut and CutNorm

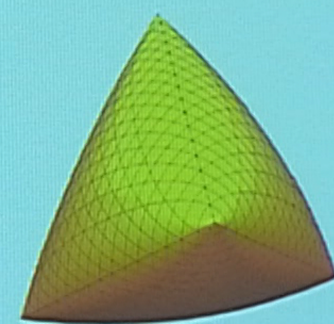
⇒ NP-hard to solve in worst case



SDP relaxation

$$\begin{array}{ll} \underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{maximize}} & \operatorname{tr}\left(\boldsymbol{A}\, \boldsymbol{x}\boldsymbol{x}^*\right) \\ \text{subject to} & \boldsymbol{x} \in \{\pm 1\}^n \end{array}$$

$$\begin{array}{ll} \underset{\boldsymbol{X} \in \mathbb{S}^n}{\text{maximize}} & \operatorname{tr}\left(\boldsymbol{A}\, \boldsymbol{X}\right) \\ \text{subject to} & \operatorname{diag}(\boldsymbol{X}) = \mathbf{1} \\ & \boldsymbol{X} \succeq \mathbf{0} \\ & \cancel{\operatorname{rank}(\boldsymbol{X}) = 1} \end{array}$$



SDP relaxation

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{maximize}} \quad & \operatorname{tr}(\boldsymbol{A}\,\boldsymbol{x}\boldsymbol{x}^*) \\ \text{subject to} \quad & \boldsymbol{x} \in \{\pm 1\}^n \end{aligned}$$

$$\begin{aligned} \underset{\boldsymbol{X}\in\mathbb{S}^n}{\text{maximize}} \quad & \operatorname{tr}(\boldsymbol{A}\,\boldsymbol{X}) \\ \text{subject to} \quad & \operatorname{diag}(\boldsymbol{X}) = \mathbf{1} \\ & \boldsymbol{X} \succeq \mathbf{0} \\ & \cancel{\operatorname{rank}(\boldsymbol{X}) = 1} \end{aligned}$$

convex relaxation:

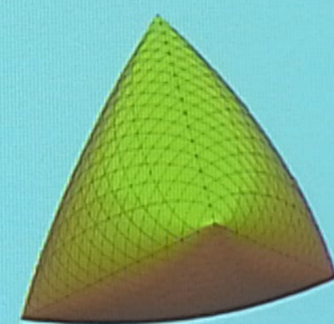
$f(\boldsymbol{X}) = \operatorname{tr}(\boldsymbol{A}\,\boldsymbol{X})$ is linear

$\boldsymbol{X} \in \mathcal{C}_1 \cap \mathcal{C}_2$ where

$\mathcal{C}_1 = \{\boldsymbol{X} : \operatorname{diag}(\boldsymbol{X}) = \mathbf{1}\}$ affine subspace

$\mathcal{C}_2 = \{\boldsymbol{X} : \boldsymbol{X} \succeq \mathbf{0}\}$ convex cone

actually a **semidefinite program** (SDP)



Fundamental problem for this talk

Caltech

$$\begin{aligned} \underset{X \in \mathbb{S}^n}{\text{maximize}} \quad & \operatorname{tr}\left(\quad A\,X \right) \\ \text{subject to} \quad & \operatorname{diag}(X) = \mathbf{1} \\ & \operatorname{tr}(X) = n, \; X \succeq \mathbf{0} \end{aligned}$$

Fundamental problem for this talk

Caltech

$$\begin{aligned}
\underset{X \in \mathbb{S}^n}{\text{maximize}} \quad & \operatorname{tr}\left(\tfrac{1}{\|A\|} A\, X\right) \\
\text{subject to} \quad & \operatorname{diag}(X) = \tfrac{1}{n}\mathbf{1} \qquad (X \in \mathcal{C}_1) \\
& \operatorname{tr}(X) = 1,\ X \succeq \mathbf{0} \qquad (X \in \mathcal{S})
\end{aligned}$$

Table of Contents

Caltech

quantum SDP speedups

Richard Küng

Strategy

Caltech

phase I: optimization problem $\Rightarrow$ feasibility problem

Strategy

Caltech

phase I: optimization problem ⇒ feasibility problem

phase II: develop quantum-inspired meta-algorithm

quantum boost: use quantum subroutines

Strategy

phase I: optimization problem ⇒ feasibility problem

phase II: develop quantum-inspired meta-algorithm

quantum boost: use quantum subroutines

inspiration: matrix multiplicative weights and mirror descent

Strategy

Caltech

phase I: optimization problem $\Rightarrow$ feasibility problem

phase II: develop quantum-inspired meta-algorithm

quantum boost: use quantum subroutines

inspiration: matrix multiplicative weights and mirror descent

Optimization $\Rightarrow$ feasibility

objective function $f(\boldsymbol{X}) = \mathrm{tr}\left(\frac{1}{\|\boldsymbol{A}\|}\boldsymbol{A}\,\boldsymbol{X}\right)$ is linear and obeys

$$|f(\boldsymbol{X})| \leq \frac{1}{\|\boldsymbol{A}\|}\|\boldsymbol{A}\|\,\|\boldsymbol{X}\|_1 = 1 \quad \text{for all} \quad \boldsymbol{X} \succeq \boldsymbol{0},\ \mathrm{tr}(\boldsymbol{X}) = 1.$$

instead of optimizing $f(\boldsymbol{X})$ directly, choose $\lambda \in [-1, 1]$ and ask:
is there a feasible $\boldsymbol{X}$ that obeys $f(\boldsymbol{X}) \leq \lambda$?

Optimization ⇒ feasibility

objective function $f(\boldsymbol{X}) = \mathrm{tr}\left(\frac{1}{\|\boldsymbol{A}\|}\boldsymbol{A}\,\boldsymbol{X}\right)$ is linear and obeys

$$|f(\boldsymbol{X})| \leq \frac{1}{\|\boldsymbol{A}\|}\|\boldsymbol{A}\|\,\|\boldsymbol{X}\|_1 = 1 \quad \text{for all} \quad \boldsymbol{X} \succeq \boldsymbol{0},\ \mathrm{tr}(\boldsymbol{X}) = 1.$$

instead of optimizing $f(\boldsymbol{X})$ directly, choose $\lambda \in [-1, 1]$ and ask:
is there a feasible $\boldsymbol{X}$ that obeys $f(\boldsymbol{X}) \leq \lambda$?

Binary search

$\mathcal{O}(2\log(1/\epsilon)) = \tilde{\mathcal{O}}(1)$ questions (with varying λ) nail down $f(\boldsymbol{X}_\sharp) \pm \epsilon$

Reformulate feasibility problem

Caltech

quantum SDP speedups

Richard Küng

task: for $\tilde{\boldsymbol{A}} = \frac{1}{\|\boldsymbol{A}\|}\boldsymbol{A}$ and $\lambda \in [-1,1]$ solve

$$
\begin{aligned}
\text{find} \quad & \boldsymbol{X} \in \mathbb{S}^n \\
\text{subject to} \quad & \mathrm{tr}\left(\tilde{\boldsymbol{A}}\,\boldsymbol{X}\right) \leq \lambda && (\boldsymbol{X} \in \mathcal{A}_\lambda) \\
& \mathrm{diag}(\boldsymbol{X}) = \tfrac{1}{n}\boldsymbol{I} && (\boldsymbol{X} \in \mathcal{D}_n) \\
& \mathrm{tr}(\boldsymbol{X}) = 1,\ \boldsymbol{X} \succeq \boldsymbol{0} && (\boldsymbol{X} \in \mathcal{S}_n)
\end{aligned}
$$

Quantum-inspired change of variables

$$\boldsymbol{X} = \rho_{\boldsymbol{H}} = \frac{\exp(-\boldsymbol{H})}{\mathrm{tr}(\exp(-\boldsymbol{H}))} \in \mathcal{S}_n \quad \text{(Gibbs state)}$$

Reformulate feasibility problem

Caltech

task: for $\tilde{\boldsymbol{A}} = \frac{1}{\|\boldsymbol{A}\|}\boldsymbol{A}$ and $\lambda \in [-1, 1]$ solve

$$
\begin{aligned}
\text{find} \quad & \boldsymbol{X} \in \mathbb{S}^n & \\
\text{subject to} \quad & \operatorname{tr}\left(\tilde{\boldsymbol{A}}\,\boldsymbol{X}\right) \leq \lambda & (\boldsymbol{X} \in \mathcal{A}_\lambda) \\
& \operatorname{diag}(\boldsymbol{X}) = \tfrac{1}{n}\boldsymbol{I} & (\boldsymbol{X} \in \mathcal{D}_n) \\
& \operatorname{tr}(\boldsymbol{X}) = 1, \ \boldsymbol{X} \succeq \boldsymbol{0} & (\boldsymbol{X} \in \mathcal{S}_n)
\end{aligned}
$$

- $\mathcal{A}_\lambda$ is half-space
- $\mathcal{D}_n$ is affine subspace
- $\mathcal{S}_n$ is the set of all density matrices

Quantum-inspired change of variables

$$\boldsymbol{X} = \rho_{\boldsymbol{H}} = \frac{\exp(-\boldsymbol{H})}{\operatorname{tr}(\exp(-\boldsymbol{H}))} \in \mathcal{S}_n \quad \text{(Gibbs state)}$$

Hamiltonian Updates

Caltech

$X \mapsto \rho_H = \frac{\exp(-H)}{\mathrm{tr}(\exp(-H))}$ automatically ensures $X \in \mathcal{S}_n$

$$
\begin{aligned}
\text{find} \quad & H \in \mathbb{S}^n \\
\text{subject to} \quad & \mathrm{tr}(\tilde{A}\, \rho_H) \leq \lambda && (\rho_H \in \mathcal{A}_\lambda) \\
& \mathrm{diag}(\rho_H) = \tfrac{1}{n} I && (\rho_H \in \mathcal{D}_n)
\end{aligned}
$$

Hamiltonian Updates:

1. start at infinite temperature, i.e. $H = 0$

[a] find separating hyperplane P and update $H \leftarrow H + \epsilon P$

Hamiltonian Updates

Caltech

$X \mapsto \rho_H = \frac{\exp(-H)}{\mathrm{tr}(\exp(-H))}$ automatically ensures $X \in \mathcal{S}_n$

$$
\begin{aligned}
\text{find} \quad & H \in \mathbb{S}^n \\
\text{subject to} \quad & \mathrm{tr}(\tilde{A}\,\rho_H) \leq \lambda && (\rho_H \in \mathcal{A}_\lambda) \\
& \mathrm{diag}(\rho_H) = \tfrac{1}{n} I && (\rho_H \in \mathcal{D}_n)
\end{aligned}
$$

Hamiltonian Updates:

1. start at infinite temperature, i.e. $H = 0$
2. check if $\rho_H \in \mathcal{A}_\lambda$ and $\rho_H \in \mathcal{D}_n$

   if true   we are done

   else   update H to penalize infeasible directions[a]

---

[a] find separating hyperplane P and update $H \leftarrow H + \epsilon P$

Hamiltonian Updates

Caltech

$\boldsymbol{X} \mapsto \rho_{\boldsymbol{H}} = \frac{\exp(-\boldsymbol{H})}{\mathrm{tr}(\exp(-\boldsymbol{H}))}$ automatically ensures $\boldsymbol{X} \in \mathcal{S}_n$

$$\begin{aligned} \text{find} \quad & \boldsymbol{H} \in \mathbb{S}^n \\ \text{subject to} \quad & \mathrm{tr}(\tilde{\boldsymbol{A}}\, \rho_{\boldsymbol{H}}) \leq \lambda & (\rho_{\boldsymbol{H}} \in \mathcal{A}_\lambda) \\ & \mathrm{diag}(\rho_{\boldsymbol{H}}) = \tfrac{1}{n}\boldsymbol{I} & (\rho_{\boldsymbol{H}} \in \mathcal{D}_n) \end{aligned}$$

Hamiltonian Updates:

1. start at infinite temperature, i.e. $\boldsymbol{H} = \boldsymbol{0}$
2. check if $\rho_{\boldsymbol{H}} \in \mathcal{A}_\lambda$ and $\rho_{\boldsymbol{H}} \in \mathcal{D}_n$

   if true we are done

   else update $\boldsymbol{H}$ to penalize infeasible directions[a]
3. loop (at most) T times

[a] find separating hyperplane $\boldsymbol{P}$ and update $\boldsymbol{H} \leftarrow \boldsymbol{H} + \epsilon \boldsymbol{P}$

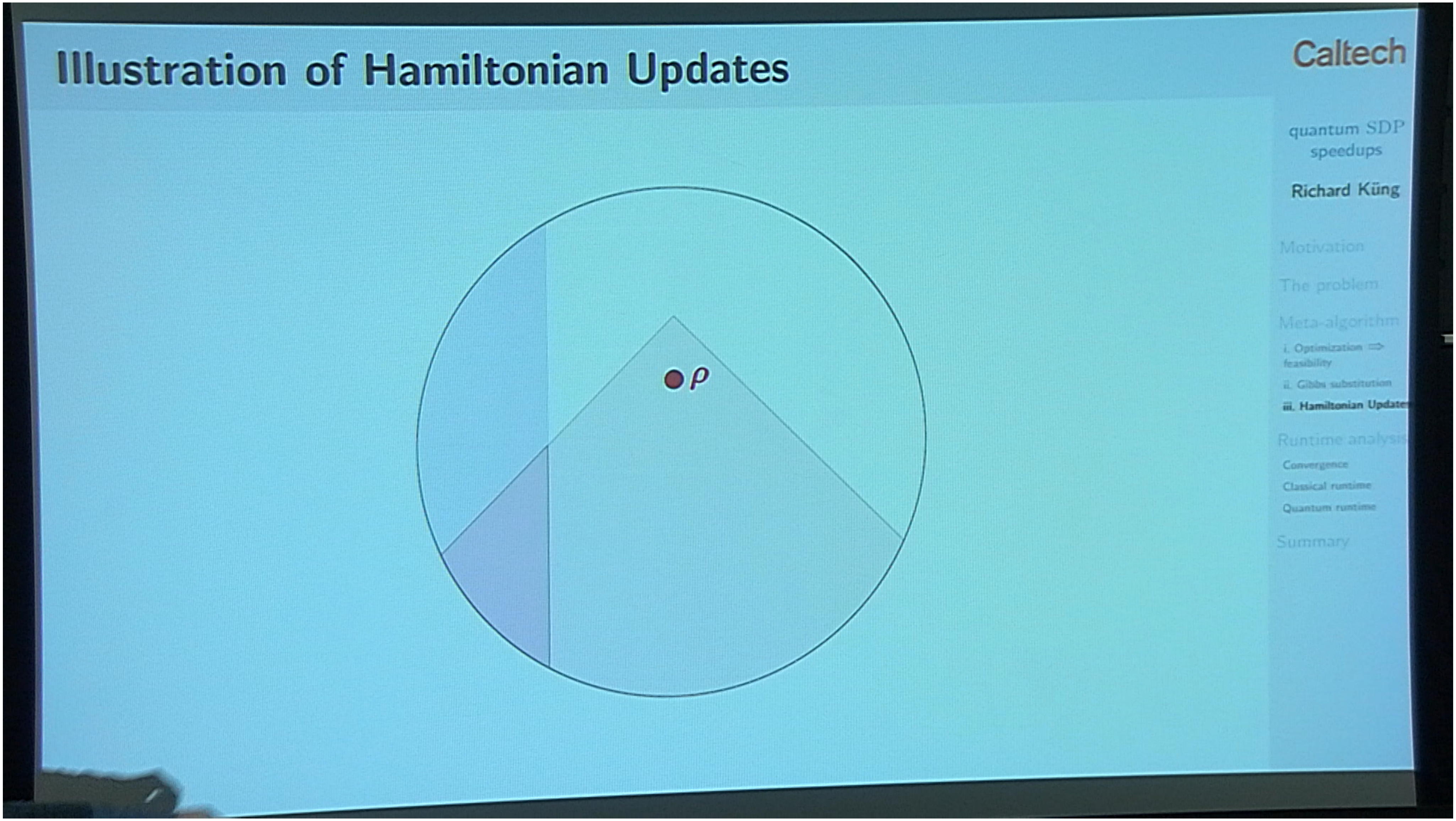
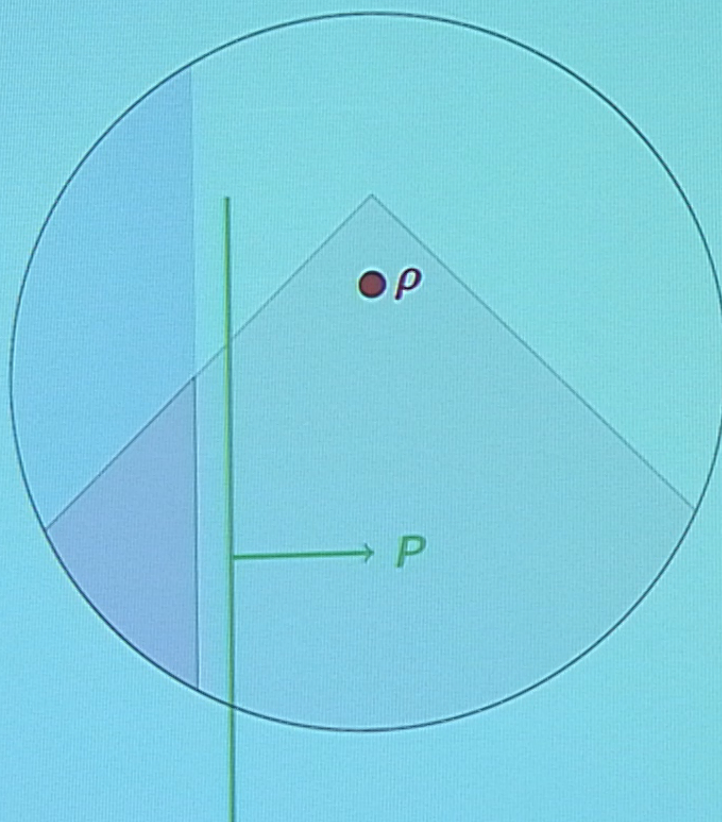


Illustration of Hamiltonian Updates
Caltech
ρ
quantum SDP speedups
Richard Küng
Motivation
The problem
Meta-algorithm
i. Optimization ⇒ feasibility
ii. Gibbs substitution
iii. Hamiltonian Updates
Runtime analysis
Convergence
Classical runtime
Quantum runtime
Summary

Illustration of Hamiltonian Updates

Caltech



quantum SDP speedups

Richard Küng

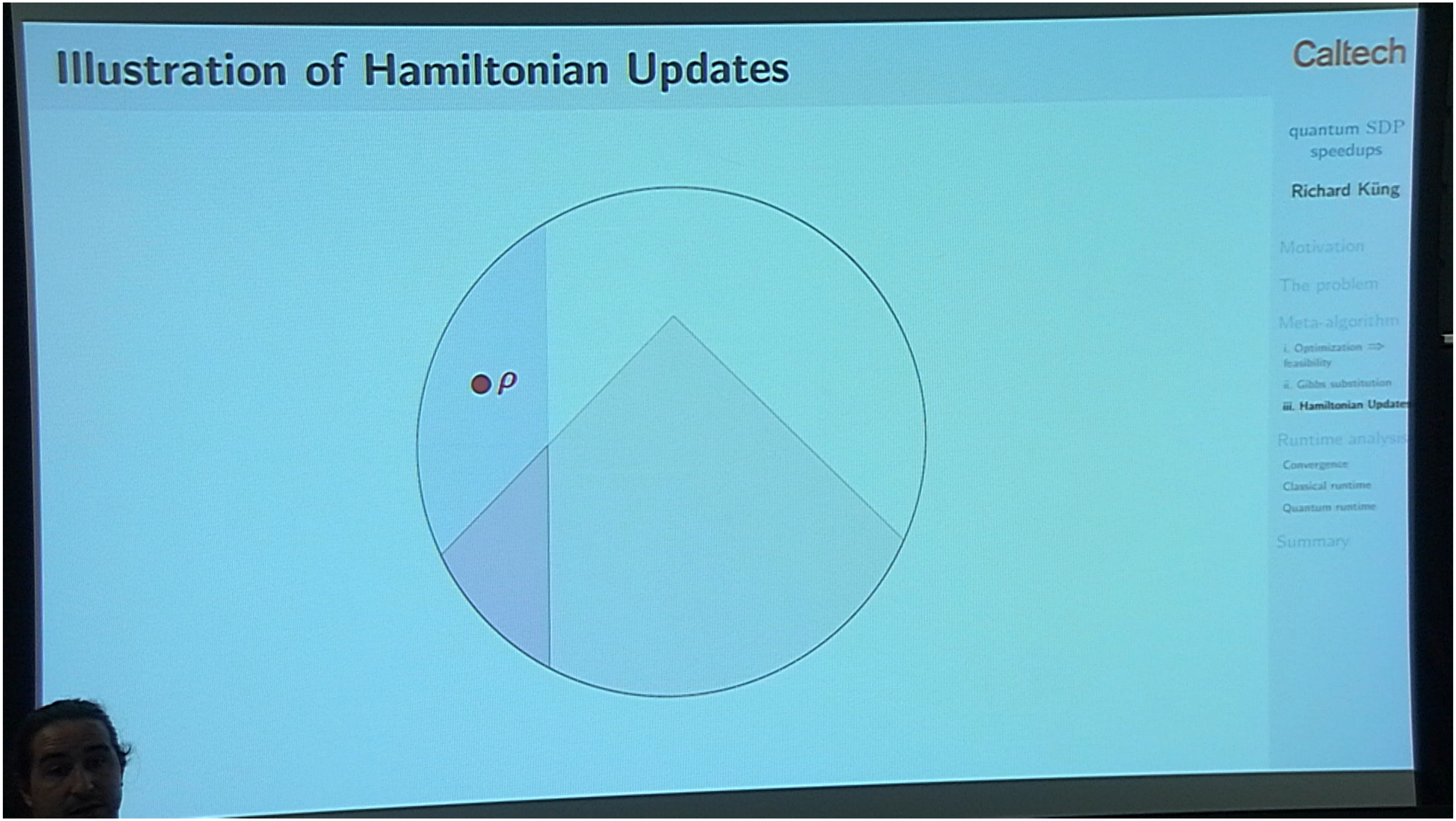


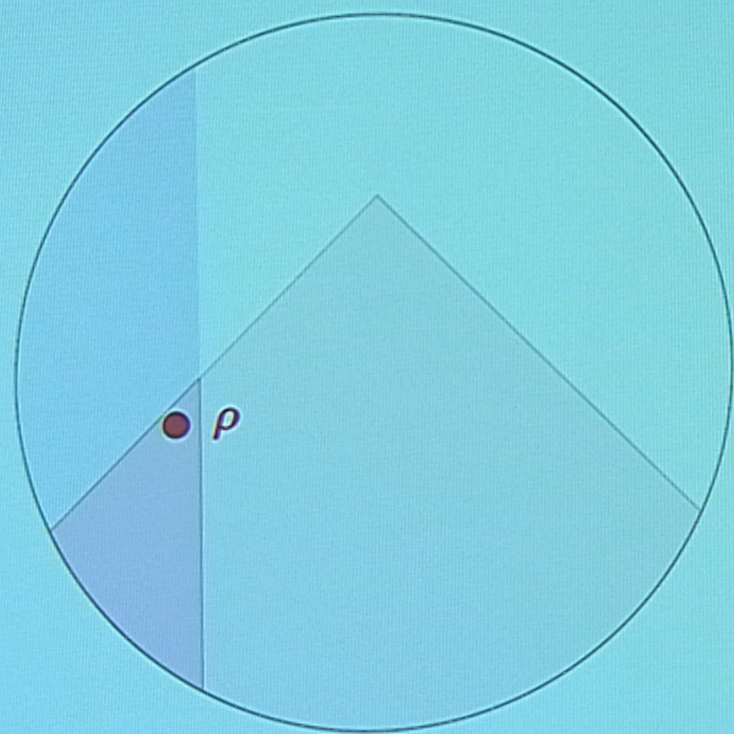
Illustration of Hamiltonian Updates
Caltech
quantum SDP speedups
Richard Küng
Motivation
The problem
Meta-algorithm
i. Optimization ⇒ feasibility
ii. Gibbs substitution
iii. Hamiltonian Updates
Runtime analysis
Convergence
Classical runtime
Quantum runtime
Summary
ρ

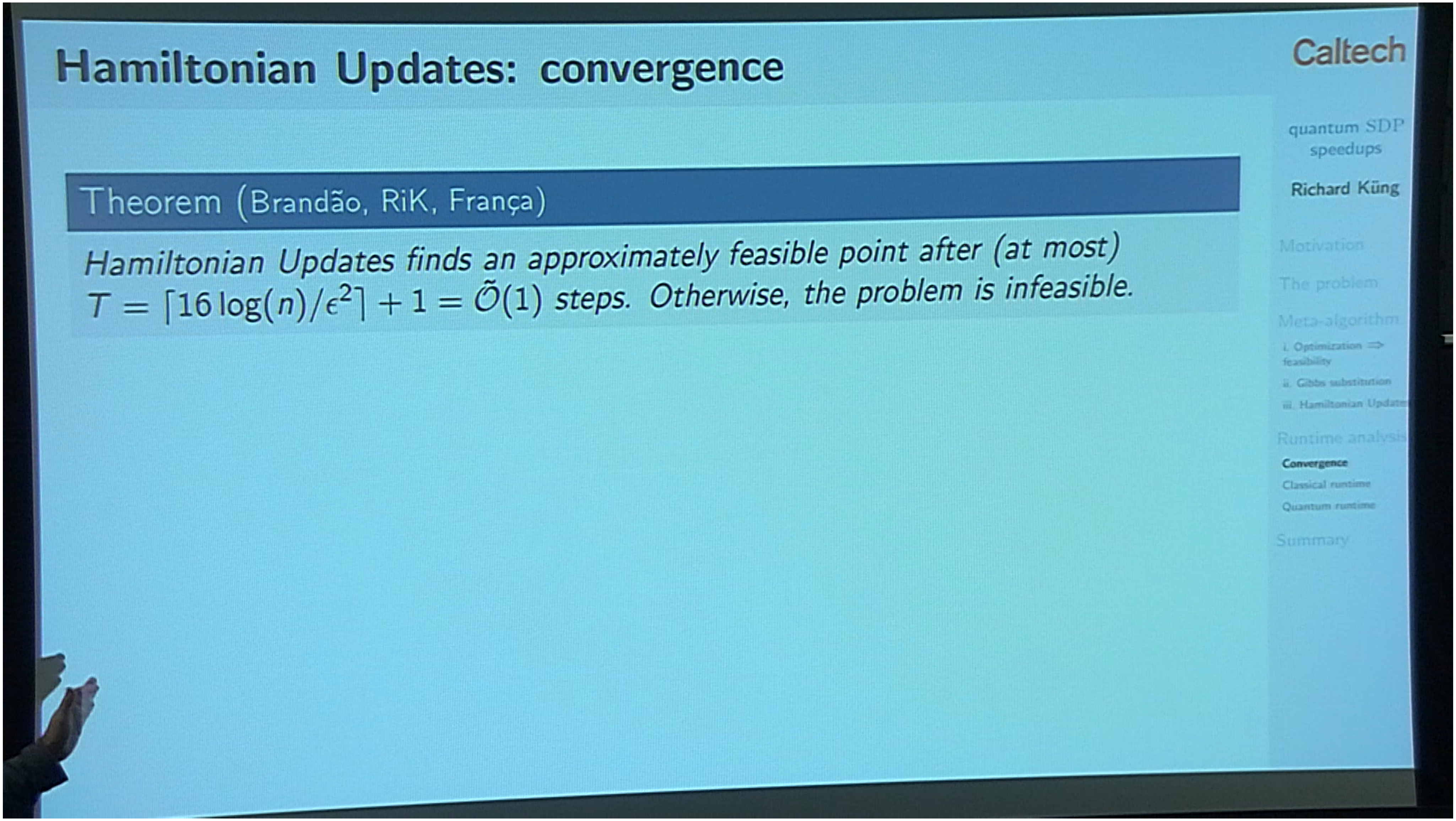
Table of Contents

Caltech

Illustration of Hamiltonian Updates

Caltech




Hamiltonian Updates: convergence
Caltech
quantum SDP speedups
Richard Küng
Motivation
The problem
Meta-algorithm
i. Optimization ⟹ feasibility
ii. Gibbs substitution
iii. Hamiltonian Updates
Runtime analysis
Convergence
Classical runtime
Quantum runtime
Summary
Theorem (Brandão, RiK, França)
Hamiltonian Updates finds an approximately feasible point after (at most) $T = \lceil 16 \log(n)/\epsilon^2 \rceil + 1 = \tilde{O}(1)$ steps. Otherwise, the problem is infeasible.

Hamiltonian Updates: convergence

Caltech

Theorem (Brandão, RiK, França)

Hamiltonian Updates finds an approximately feasible point after (at most) $T = \lceil 16 \log(n)/\epsilon^2 \rceil + 1 = \tilde{O}(1)$ *steps. Otherwise, the problem is infeasible.*

proof idea:

- relative entropy between $\rho_0 = \frac{1}{n} I$ and *any* feasible point ρ^* is $\leq \log(n)$

Hamiltonian Updates: convergence

Theorem (Brandão, RiK, França)

Hamiltonian Updates finds an approximately feasible point after (at most) $T = \lceil 16 \log(n)/\epsilon^2 \rceil + 1 = \tilde{O}(1)$ steps. Otherwise, the problem is infeasible.

proof idea:

- relative entropy between $\rho_0 = \frac{1}{n} I$ and *any* feasible point ρ^* is $\leq \log(n)$
- show that each iteration makes constant progress in relative entropy:

$$S(\rho^* \| \rho_{t+1}) - S(\rho^* \| \rho_t) \leq -\frac{\epsilon^2}{16}$$

$\Rightarrow$ convergence after (at most) T steps, or $S(\rho^* \| \rho_T) < 0$

Hamiltonian Updates: computational cost

Caltech

- Hamiltonian Updates solves feasibility problem in $\mathcal{O}(\log(n)/\epsilon^2) = \tilde{\mathcal{O}}(1)$ steps

Hamiltonian Updates: computational cost

- Hamiltonian Updates solves feasibility problem in $\mathcal{O}(\log(n)/\epsilon^2) = \tilde{\mathcal{O}}(1)$ steps
- each step requires three subroutines:
  - (i) compute $\rho_H = \frac{\exp(-H)}{\mathrm{tr}(\exp(-H))}$
  - (ii) $\rho_H \in \mathcal{A}_\lambda$: check $\mathrm{tr}(\tilde{A}\,\rho_H) \leq \lambda$; output $P = \tilde{A}$
  - (iii) $\rho_H \in \mathcal{D}_n$: check $\mathrm{diag}(\rho_H) = \frac{1}{n}\mathbf{1}$; output $P = \sum_i \mathbb{I}\left\{\langle e_i, \rho_H e_i\rangle > \frac{1}{n}\right\} e_i e_i^t$

Hamiltonian Updates: computational cost

- Hamiltonian Updates solves feasibility problem in $\mathcal{O}(\log(n)/\epsilon^2) = \tilde{\mathcal{O}}(1)$ steps
- each step requires three subroutines:
  - (i) compute $\rho_H = \frac{\exp(-H)}{\mathrm{tr}(\exp(-H))}$
  - (ii) $\rho_H \in \mathcal{A}_\lambda$: check $\mathrm{tr}(\tilde{A}\,\rho_H) \leq \lambda$; output $P = \tilde{A}$
  - (iii) $\rho_H \in \mathcal{D}_n$: check $\mathrm{diag}(\rho_H) = \frac{1}{n}\mathbf{1}$; output $P = \sum_i \mathbb{I}\left\{\langle e_i, \rho_H e_i\rangle > \frac{1}{n}\right\} e_i e_i^t$
- naive cost:
  - (i) $\mathcal{O}(n^3)$
  - (ii) $\mathcal{O}(ns)$ $\quad s = (\mathrm{row})\mathrm{sparsity}(\tilde{A})$
  - (iii) $\mathcal{O}(n)$

Hamiltonian Updates: computational cost

Caltech

quantum SDP speedups

Richard Küng

Motivation

The problem

Meta-algorithm

i. Optimization ⟹ feasibility

ii. Gibbs substitution

iii. Hamiltonian Updates

Runtime analysis

Convergence

Classical runtime

Quantum runtime

Summary

- Hamiltonian Updates solves feasibility problem in $\mathcal{O}(\log(n)/\epsilon^2) = \tilde{\mathcal{O}}(1)$ steps
- each step requires three subroutines:
  - (i) compute $\rho_H = \frac{\exp(-H)}{\mathrm{tr}(\exp(-H))}$
  - (ii) $\rho_H \in \mathcal{A}_\lambda$: check $\mathrm{tr}(\tilde{A}\,\rho_H) \leq \lambda$; output $P = \tilde{A}$
  - (iii) $\rho_H \in \mathcal{D}_n$: check $\mathrm{diag}(\rho_H) = \frac{1}{n}\mathbf{1}$; output $P = \sum_i \mathbb{I}\left\{\langle \mathbf{e}_i, \rho_H \mathbf{e}_i\rangle > \frac{1}{n}\right\} \mathbf{e}_i \mathbf{e}_i^t$
- naive cost:
  - (i) $\mathcal{O}(n^3)$
  - (ii) $\mathcal{O}(ns)$ $\quad s = \text{(row)sparsity}(\tilde{A})$
  - (iii) $\mathcal{O}(n)$
- naive total cost: $\tilde{\mathcal{O}}(n^4 s)$ (not very impressive yet)

Hamiltonian Updates: classical implementation

Caltech

quantum SDP speedups

Richard Küng

fact: Hamiltonian updates is designed to be robust

⇒ implementing subroutines up to accuracy ϵ still yields an approximately feasible solution (and correctly flags infeasibility)

Hamiltonian Updates: classical implementation

Caltech

fact: Hamiltonian updates is designed to be robust

$\Rightarrow$ implementing subroutines up to accuracy ϵ still yields an approximately feasible solution (and correctly flags infeasibility)

classical boost: $\exp(-\boldsymbol{H}) \simeq \sum_{k=0}^{\ell} \frac{\boldsymbol{H}^k}{k!}$, $\ell = \mathcal{O}(\log(n)/\epsilon) = \tilde{\mathcal{O}}(1)$

Hamiltonian Updates: classical implementation

fact: Hamiltonian updates is designed to be robust

⇒ implementing subroutines up to accuracy ϵ still yields an approximately feasible solution (and correctly flags infeasibility)

classical boost: $\exp(-\boldsymbol{H}) \simeq \sum_{k=0}^{\ell} \frac{\boldsymbol{H}^k}{k!}$, $\ell = \mathcal{O}(\log(n)/\epsilon) = \tilde{\mathcal{O}}(1)$

Theorem (Brandão, RiK, França; 2019)

Hamiltonian Updates approximately solves binary quadratic SDP relaxations in classical runtime $\mathcal{O}\left(n^2 s \log(n)/\epsilon^{12}\right) = \tilde{\mathcal{O}}(n^2 s)$, *where* $s = (row)sparsity(\boldsymbol{A})$.

$$\begin{aligned} \text{maximize} \quad & \operatorname{tr}\left(\frac{1}{\|\boldsymbol{A}\|}\boldsymbol{A}\,\boldsymbol{X}\right) \\ \text{subject to} \quad & \operatorname{diag}(\boldsymbol{X}) = \mathbf{1} \\ & \boldsymbol{X} \succeq \mathbf{0}. \end{aligned}$$

Hamiltonian Updates: classical implementation

fact: Hamiltonian updates is designed to be robust

$\Rightarrow$ implementing subroutines up to accuracy ϵ still yields an approximately feasible solution (and correctly flags infeasibility)

classical boost: $\exp(-\boldsymbol{H}) \simeq \sum_{k=0}^{\ell} \frac{\boldsymbol{H}^k}{k!}$, $\ell = \mathcal{O}(\log(n)/\epsilon) = \tilde{\mathcal{O}}(1)$

Theorem (Brandão, RiK, França; 2019)

Hamiltonian Updates approximately solves binary quadratic SDP relaxations in classical runtime $\mathcal{O}\left(n^2 s \log(n)/\epsilon^{12}\right) = \tilde{\mathcal{O}}(n^2 s)$, *where* $s = (row)sparsity(\boldsymbol{A})$.

$$\begin{aligned} \text{maximize} \quad & \mathrm{tr}\left(\tfrac{1}{\|\boldsymbol{A}\|}\boldsymbol{A}\,\boldsymbol{X}\right) \\ \text{subject to} \quad & \mathrm{diag}(\boldsymbol{X}) = \mathbf{1} \\ & \boldsymbol{X} \succeq \mathbf{0}. \end{aligned}$$

1. best existing general algorithm: $\tilde{\mathcal{O}}(n^{2.5} s)$
2. approx. discrepancy: $\epsilon n \|\boldsymbol{A}\|$ vs. $\epsilon \|\boldsymbol{A}\|_{\ell_1}$
3. favorable for generic problem instances
4. no speedup for MaxCut

Hamiltonian Updates: classical implementation

fact: Hamiltonian updates is designed to be robust

⇒ implementing subroutines up to accuracy ϵ still yields an approximately feasible solution (and correctly flags infeasibility)

classical boost: $\exp(-\boldsymbol{H}) \simeq \sum_{k=0}^{\ell} \frac{\boldsymbol{H}^k}{k!}$, $\ell = \mathcal{O}(\log(n)/\epsilon) = \tilde{\mathcal{O}}(1)$

Theorem (Brandão, RiK, França; 2019)

Hamiltonian Updates approximately solves binary quadratic SDP relaxations in classical runtime $\mathcal{O}\left(n^2 s \log(n)/\epsilon^{12}\right) = \tilde{\mathcal{O}}(n^2 s)$, *where* $s = (row)sparsity(\boldsymbol{A})$.

$$\begin{aligned} \text{maximize} \quad & \operatorname{tr}\left(\tfrac{1}{\|\boldsymbol{A}\|}\boldsymbol{A}\,\boldsymbol{X}\right) \\ \text{subject to} \quad & \operatorname{diag}(\boldsymbol{X}) = \mathbf{1} \\ & \boldsymbol{X} \succeq \mathbf{0}. \end{aligned}$$

1. best existing general algorithm: $\tilde{\mathcal{O}}(n^{2.5} s)$
2. approx. discrepancy: $\epsilon n \|\boldsymbol{A}\|$ vs. $\epsilon \|\boldsymbol{A}\|_{\ell_1}$
3. favorable for generic problem instances
4. no speedup for MaxCut

Hamiltonian updates: quantum implementation

Caltech

classical bottleneck: compute Gibbs states $\rho_H = \frac{\exp(-H)}{\mathrm{tr}(\exp(-H))}$

Hamiltonian updates: quantum implementation

Caltech

quantum SDP speedups

Richard Küng

classical bottleneck: compute Gibbs states $\rho_H = \frac{\exp(-H)}{\mathrm{tr}(\exp(-H))}$

quantum speedup:

prepare copies of ρ_H on quantum computer — $\tilde{\mathcal{O}}(\sqrt{n}s s^{o(1)})$

estimate $\mathrm{tr}(\tilde{A}\,\rho_H)$ via phase estimation — $\mathcal{O}(1/\epsilon^2)$ copies

estimate $\mathrm{diag}(\rho_H)$ via computational basis measurements — $\mathcal{O}(n/\epsilon^2)$ copies

Theorem (Brandão, RiK, França; 2019)

Hamiltonian Updates approximately solves binary quadratic SDP relaxations in quantum runtime $\tilde{\mathcal{O}}(n^{1.5}(\sqrt{s})^{1+o(1)})$.

Hamiltonian updates: quantum implementation

Caltech

classical bottleneck: compute Gibbs states $\rho_H = \frac{\exp(-\boldsymbol{H})}{\mathrm{tr}(\exp(-\boldsymbol{H}))}$

quantum speedup:

- prepare copies of ρ_H on quantum computer — $\tilde{\mathcal{O}}(\sqrt{n}s s^{o(1)})$
- estimate $\mathrm{tr}(\tilde{\boldsymbol{A}}\,\rho_H)$ via phase estimation — $\mathcal{O}(1/\epsilon^2)$ copies
- estimate $\mathrm{diag}(\rho_H)$ via computational basis measurements — $\mathcal{O}(n/\epsilon^2)$ copies

Theorem (Brandão, RiK, França; 2019)

Hamiltonian Updates approximately solves binary quadratic SDP relaxations in quantum runtime $\tilde{\mathcal{O}}(n^{1.5}(\sqrt{s})^{1+o(1)})$.

1. first quantum speedup for **important** SDP class
2. beats classical runtimes $\tilde{\mathcal{O}}(n^2 s)$ and $\tilde{\mathcal{O}}(n^{2.5} s)$
3. classical access to (approx.) optimal Hamiltonian ⇒ data processing

Details about quantum subroutine

Caltech

important design feature: Hamiltonians are very structured:
$\boldsymbol{H} = \alpha \tilde{\boldsymbol{A}} + \beta \boldsymbol{D}$, $\alpha, \beta = \mathcal{O}(\log(n)/\epsilon)$

Details about quantum subroutine

Caltech

important design feature: Hamiltonians are very structured:
$\boldsymbol{H} = \alpha \tilde{\boldsymbol{A}} + \beta \boldsymbol{D}$, $\alpha, \beta = \mathcal{O}(\log(n)/\epsilon)$

1. use [Poulin, Wojcan; 2009] to reduce task of preparing ρ_H to simulating time evolution ($\mathcal{O}(\sqrt{n})$ invocations)
2. use [Childs, Wiebe; 2012] to split up time evolution (negligible overhead)
3. [Low; 2019]: implementing $\exp(it\alpha\tilde{\boldsymbol{A}})$ costs $\tilde{\mathcal{O}}(\sqrt{s}^{1+o(1)})$
4. [Prakash; 2014] implementing $\exp(it\beta\boldsymbol{D})$ with quantum RAM costs $\tilde{\mathcal{O}}(n)$

$\Rightarrow$ total cost: $\tilde{\mathcal{O}}(n^{1.5}\sqrt{s}^{1+o(1)})$

Table of Contents

Caltech

Conclusion

Caltech

we established speedups for important problem class:

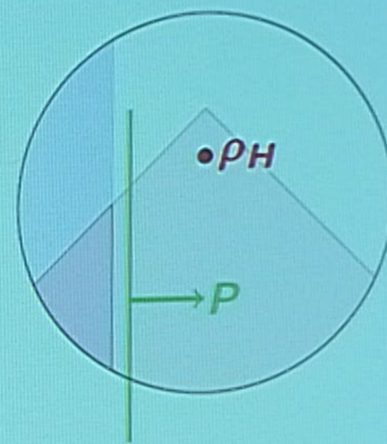
$$\begin{aligned} \underset{\boldsymbol{X} \in \mathbb{S}^n}{\text{maximize}} \quad & \mathrm{tr}(\boldsymbol{A}\,\boldsymbol{X}) \\ \text{subject to} \quad & \mathrm{diag}(\boldsymbol{X}) = \tfrac{1}{n}\mathbf{1} \\ & \mathrm{tr}(\boldsymbol{X}) = 1,\ \boldsymbol{X} \succeq \mathbf{0} \end{aligned}$$

Conclusion

Caltech

we established speedups for important problem class:

$$\begin{aligned} \underset{X \in \mathbb{S}^n}{\text{maximize}} \quad & \operatorname{tr}(A\,X) \\ \text{subject to} \quad & \operatorname{diag}(X) = \tfrac{1}{n}\mathbf{1} \\ & \operatorname{tr}(X) = 1,\; X \succeq \mathbf{0} \end{aligned}$$



our strategy:

(i) replace optimization by a sequence of feasibility problems

(ii) change of variables: $X \leftarrow \rho_H = \frac{\exp(-H)}{\operatorname{tr}(\exp(-H))}$

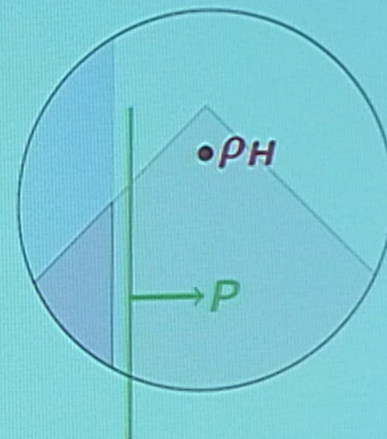
(iii) iteratively penalize infeasible directions by Hamiltonian Updates $H \leftarrow H + \epsilon P$

(iv) boost runtime by preparing each ρ_H on quantum computer

Conclusion

we established speedups for important problem class:

$$\begin{aligned} \underset{\boldsymbol{X} \in \mathbb{S}^n}{\text{maximize}} \quad & \operatorname{tr}(\boldsymbol{A}\,\boldsymbol{X}) \\ \text{subject to} \quad & \operatorname{diag}(\boldsymbol{X}) = \tfrac{1}{n}\mathbf{1} \\ & \operatorname{tr}(\boldsymbol{X}) = 1,\ \boldsymbol{X} \succeq \mathbf{0} \end{aligned}$$



our strategy:

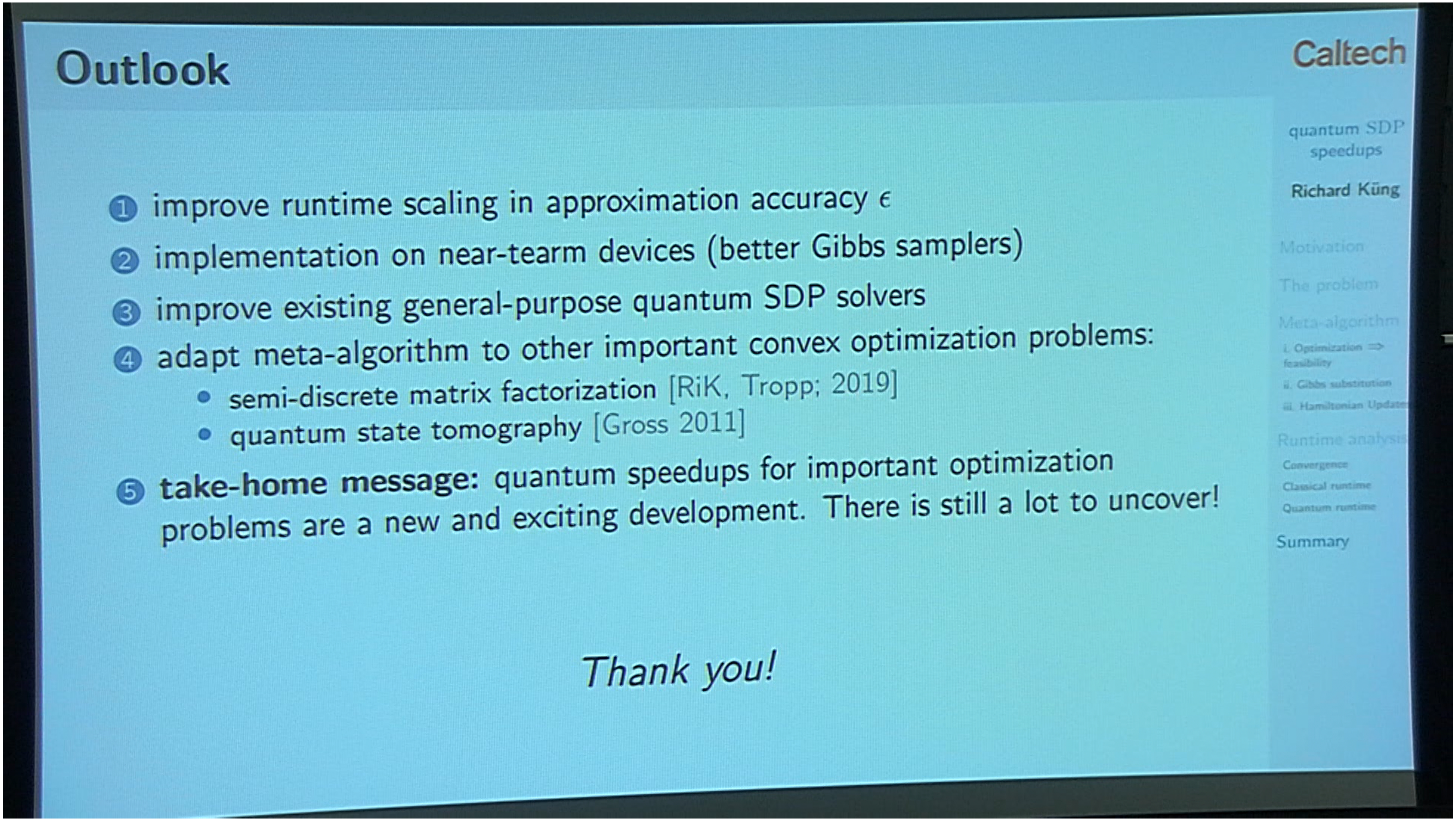
(i) replace optimization by a sequence of feasibility problems

(ii) change of variables: $\boldsymbol{X} \leftarrow \rho_{\boldsymbol{H}} = \frac{\exp(-\boldsymbol{H})}{\operatorname{tr}(\exp(-\boldsymbol{H}))}$

(iii) iteratively penalize infeasible directions by Hamiltonian Updates $\boldsymbol{H} \leftarrow \boldsymbol{H} + \epsilon \boldsymbol{P}$

(iv) boost runtime by preparing each $\rho_{\boldsymbol{H}}$ on quantum computer

our result: we obtain approximate solutions faster than existing approaches: $\tilde{O}(n^2 s)$ (classical) and $\tilde{O}(n^{1.5}\sqrt{s}^{1+o(1)})$ (quantum) vs. $\tilde{O}(n^{2.5} s)$ (classical)


Outlook
Caltech
quantum SDP speedups
Richard Küng
Motivation
The problem
Meta-algorithm
i. Optimization ⇒ feasibility
ii. Gibbs substitution
Runtime analysis
Convergence
Classical runtime
Quantum runtime
Summary
1 improve runtime scaling in approximation accuracy ϵ
2 implementation on near-tearm devices (better Gibbs samplers)
3 improve existing general-purpose quantum SDP solvers
4 adapt meta-algorithm to other important convex optimization problems:
• semi-discrete matrix factorization [RiK, Tropp; 2019]
• quantum state tomography [Gross 2011]
5 take-home message: quantum speedups for important optimization problems are a new and exciting development. There is still a lot to uncover!
Thank you!