"High-Precision" Quantum Algorithms

Somma, Rolando D.

Theoretical Division
Los Alamos National Laboratory, NM, US

February 17th, 2015
ASCR Workshop on Quantum Computing for Science
Washington, DC

Collaborators: A. Childs (UMD), R. Kothari (MIT)

Related work: D. Berry, A.M. Childs, R. Cleve, R. Kothari, R.D. Somma,
arXiv 1312.1414 (STOC), arXiv 1412.4687 (PRL)

LA-UR-15-20956

The problem

Reduce L, the number of gates in quantum circuits

- What to do with a small-to-mid size quantum computer?
- Quantum simulations?
- Solving linear systems of equations?
- Solving differential equations?
- what else?

Observation 1: If we do not have a way to synthesize circuits, L can be ridiculously large [1]

[1] D. Wecker, et.al., Gate count estimates for performing quantum chemistry on small quantum computers, Phys. Rev. A **90**, 022305 (2014).

The problem

Reduce L, the number of gates in quantum circuits

- What to do with a small-to-mid size quantum computer?
- Quantum simulations?
- Solving linear systems of equations?
- Solving differential equations?
- what else?

Observation 1: If we do not have a way to synthesize circuits, L can be ridiculously large [1]

[1] D. Wecker, et.al., Gate count estimates for performing quantum chemistry on small quantum computers, Phys. Rev. A **90**, 022305 (2014).

The problem

Reduce L, the number of gates in quantum circuits

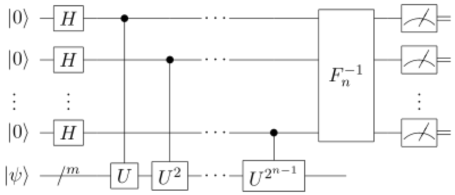
- What to do with a small-to-mid size quantum computer?
- Quantum simulations?
- Solving linear systems of equations?
- Solving differential equations?
- what else?

Observation 1: If we do not have a way to synthesize circuits, L can be ridiculously large [1]

[1] D. Wecker, et.al., Gate count estimates for performing quantum chemistry on small quantum computers, Phys. Rev. A **90**, 022305 (2014).

The problem

In many quantum algorithms, L has a poor scaling with precision ($\epsilon \ll 1$)



$$\longleftrightarrow$$

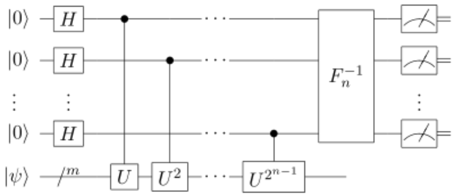
$$L \sim \text{poly}(1/\epsilon)$$

- Simulating Hamiltonian dynamics using Trotte-Suzuki decompositions
- Algorithms that use phase estimation as a subroutine

Examples: computation of physical properties, applying inverse of matrices, and more

The problem

In many quantum algorithms, L has a poor scaling with precision ($\epsilon \ll 1$)



$$\longleftrightarrow$$

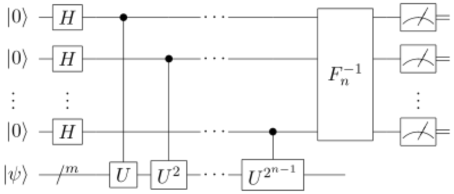
$$L \sim \text{poly}(1/\epsilon)$$

- Simulating Hamiltonian dynamics using Trotte-Suzuki decompositions
- Algorithms that use phase estimation as a subroutine

Examples: computation of physical properties, applying inverse of matrices, and more

The problem

In many quantum algorithms, L has a poor scaling with precision ($\epsilon \ll 1$)



$$\longleftrightarrow$$
$$L \sim \text{poly}(1/\epsilon)$$

- Simulating Hamiltonian dynamics using Trotte-Suzuki decompositions
- Algorithms that use phase estimation as a subroutine

Examples: computation of physical properties, applying inverse of matrices, and more

Our goal

Develop new quantum circuits where $L \rightarrow \mathrm{polylog}(1/\epsilon)$:
“high-precision" quantum algorithms

- Simulating physical systems: real and imaginary time evolutions
- Linear algebra problems: solving linear systems of equations, solving differential equations, etc.

Related work

- Simulating Hamiltonian dynamics [Berry, Childs, Cleve, Kothari, **RS**]
- Solovey-Kitaev to approximate simple unitary operations with gates from a universal set
- More efficient synthesis of quantum circuits [groups at Microsoft, Harvard, IQC,...] (Krysta’s talk)

Our goal

Develop new quantum circuits where $L \to \mathrm{polylog}(1/\epsilon)$: "high-precision" quantum algorithms

- Simulating physical systems: real and imaginary time evolutions
- Linear algebra problems: solving linear systems of equations, solving differential equations, etc.

Related work

- Simulating Hamiltonian dynamics [Berry, Childs, Cleve, Kothari, **RS**]
- Solovey-Kitaev to approximate simple unitary operations with gates from a universal set
- More efficient synthesis of quantum circuits [groups at Microsoft, Harvard, IQC,...] (Krysta's talk)

Our goal

Develop new quantum circuits where $L \to \mathrm{polylog}(1/\epsilon)$: "high-precision" quantum algorithms

- Simulating physical systems: real and imaginary time evolutions
- Linear algebra problems: solving linear systems of equations, solving differential equations, etc.

Related work

- Simulating Hamiltonian dynamics [Berry, Childs, Cleve, Kothari, **RS**]
- Solovey-Kitaev to approximate simple unitary operations with gates from a universal set
- More efficient synthesis of quantum circuits [groups at Microsoft, Harvard, IQC,...] (Krysta's talk)

Our goal

Develop new quantum circuits where $L \rightarrow \mathrm{polylog}(1/\epsilon)$:
“high-precision" quantum algorithms

- Simulating physical systems: real and imaginary time evolutions
- Linear algebra problems: solving linear systems of equations, solving differential equations, etc.

Related work

- Simulating Hamiltonian dynamics [Berry, Childs, Cleve, Kothari, **RS**]
- Solovey-Kitaev to approximate simple unitary operations with gates from a universal set
- More efficient synthesis of quantum circuits [groups at Microsoft, Harvard, IQC,...] (Krysta’s talk)

Useful tools: Linear combinations of unitaries

Goal: prepare $A|\psi\rangle$, $\|A\| \le 1$.

Theorem (1)

Let $A = \sum_{j=0}^{m-1} \beta_j V_j$, where the V_j's are unitary, $\beta_j > 0$, $\sum_{j=0}^{m-1} \beta_j \le 1$. Assume we have access to a unitary $\bar{V} = \sum_{j=0}^{m-1} V_j \otimes |j\rangle\langle j|$. Then, we can simulate A on a quantum computer with one use of $\bar{V}$.

Proof.

- $|\psi\rangle |0\rangle$
- $|\psi\rangle \sum_{j=0}^{m} \sqrt{\beta_j}\, |j\rangle$, with $\beta_m = 1 - \sum_{j=0}^{m-1} \beta_j$
- $\sum_{j=0}^{m} \sqrt{\beta_j}\, V_j |\psi\rangle |j\rangle$
- $\sum_{j=0}^{m-1} \beta_j V_j |\psi\rangle |0\rangle + |\phi^\perp\rangle = A|\psi\rangle |0\rangle + |\phi^\perp\rangle$

$\square$

Useful tools: Linear combinations of unitaries

Goal: prepare $A|\psi\rangle$, $\|A\| \leq 1$.

Theorem (1)

Let $A = \sum_{j=0}^{m-1} \beta_j V_j$, where the V_j's are unitary, $\beta_j > 0$, $\sum_{j=0}^{m-1} \beta_j \leq 1$. Assume we have access to a unitary $\bar{V} = \sum_{j=0}^{m-1} V_j \otimes |j\rangle\langle j|$. Then, we can simulate A on a quantum computer with one use of $\bar{V}$.

Proof.

- $|\psi\rangle|0\rangle$
- $|\psi\rangle \sum_{j=0}^{m} \sqrt{\beta_j}\,|j\rangle$, with $\beta_m = 1 - \sum_{j=0}^{m-1} \beta_j$
- $\sum_{j=0}^{m} \sqrt{\beta_j}\, V_j |\psi\rangle |j\rangle$
- $\sum_{j=0}^{m-1} \beta_j V_j |\psi\rangle |0\rangle + |\phi^\perp\rangle = A|\psi\rangle|0\rangle + |\phi^\perp\rangle$

$\square$

Useful tools: Linear combinations of unitaries

Goal: prepare $A|\psi\rangle$, $\|A\| \leq 1$.

Theorem (1)

Let $A = \sum_{j=0}^{m-1} \beta_j V_j$, where the V_j's are unitary, $\beta_j > 0$, $\sum_{j=0}^{m-1} \beta_j \leq 1$. Assume we have access to a unitary $\bar{V} = \sum_{j=0}^{m-1} V_j \otimes |j\rangle\langle j|$. Then, we can simulate A on a quantum computer with one use of $\bar{V}$.

Proof.

- $|\psi\rangle|0\rangle$
- $|\psi\rangle \sum_{j=0}^{m} \sqrt{\beta_j}|j\rangle$, with $\beta_m = 1 - \sum_{j=0}^{m-1} \beta_j$
- $\sum_{j=0}^{m} \sqrt{\beta_j} V_j |\psi\rangle |j\rangle$
- $\sum_{j=0}^{m-1} \beta_j V_j |\psi\rangle|0\rangle + |\phi^\perp\rangle = A|\psi\rangle|0\rangle + |\phi^\perp\rangle$

$\square$

Useful tools: Linear combinations of unitaries

Goal: prepare $A|\psi\rangle$, $\|A\| \leq 1$.

Theorem (1)

Let $A = \sum_{j=0}^{m-1} \beta_j V_j$, where the V_j's are unitary, $\beta_j > 0$, $\sum_{j=0}^{m-1} \beta_j \leq 1$. Assume we have access to a unitary $\bar{V} = \sum_{j=0}^{m-1} V_j \otimes |j\rangle\langle j|$. Then, we can simulate A on a quantum computer with one use of $\bar{V}$.

Proof.

- $|\psi\rangle|0\rangle$
- $|\psi\rangle \sum_{j=0}^{m} \sqrt{\beta_j}\,|j\rangle$, with $\beta_m = 1 - \sum_{j=0}^{m-1} \beta_j$
- $\sum_{j=0}^{m} \sqrt{\beta_j} V_j |\psi\rangle |j\rangle$
- $\sum_{j=0}^{m-1} \beta_j V_j |\psi\rangle |0\rangle + |\phi^\perp\rangle = A|\psi\rangle|0\rangle + |\phi^\perp\rangle$

□

Useful tools: Linear combinations of unitaries

Goal: prepare $A|\psi\rangle$, $\|A\| \leq 1$.

Theorem (1)

Let $A = \sum_{j=0}^{m-1} \beta_j V_j$, where the V_j's are unitary, $\beta_j > 0$, $\sum_{j=0}^{m-1} \beta_j \leq 1$. Assume we have access to a unitary $\bar{V} = \sum_{j=0}^{m-1} V_j \otimes |j\rangle\langle j|$. Then, we can simulate A on a quantum computer with one use of $\bar{V}$.

Proof.

- $|\psi\rangle|0\rangle$
- $|\psi\rangle \sum_{j=0}^{m} \sqrt{\beta_j}|j\rangle$, with $\beta_m = 1 - \sum_{j=0}^{m-1} \beta_j$
- $\sum_{j=0}^{m} \sqrt{\beta_j} V_j |\psi\rangle |j\rangle$
- $\sum_{j=0}^{m-1} \beta_j V_j |\psi\rangle|0\rangle + |\phi^{\perp}\rangle = A|\psi\rangle|0\rangle + |\phi^{\perp}\rangle$

□

Useful tools: Linear combinations of unitaries

Goal: prepare $A|\psi\rangle$, $\|A\| \leq 1$.

Theorem (1)

Let $A = \sum_{j=0}^{m-1} \beta_j V_j$, where the V_j's are unitary, $\beta_j > 0$, $\sum_{j=0}^{m-1} \beta_j \leq 1$. Assume we have access to a unitary $\bar{V} = \sum_{j=0}^{m-1} V_j \otimes |j\rangle\langle j|$. Then, we can simulate A on a quantum computer with one use of $\bar{V}$.

Proof.

- $|\psi\rangle|0\rangle$
- $|\psi\rangle \sum_{j=0}^{m} \sqrt{\beta_j}\,|j\rangle$, with $\beta_m = 1 - \sum_{j=0}^{m-1} \beta_j$
- $\sum_{j=0}^{m} \sqrt{\beta_j}\, V_j |\psi\rangle |j\rangle$
- $\sum_{j=0}^{m-1} \beta_j V_j |\psi\rangle |0\rangle + |\phi^\perp\rangle = A|\psi\rangle|0\rangle + |\phi^\perp\rangle$

□

Observations

- Observation 2: In applications, we will need to consider the implementation cost of $\bar{V}$ in terms of two-qubit gates.

- Observation 3: In applications, the exact decomposition of A in terms of unitaries may require too many terms. We can then approximate A, within some ϵ, using a different decomposition with fewer terms. Various approximation methods will be relevant here.

Observations

- Observation 2: In applications, we will need to consider the implementation cost of $\bar{V}$ in terms of two-qubit gates.

- Observation 3: In applications, the exact decomposition of A in terms of unitaries may require too many terms. We can then approximate A, within some ϵ, using a different decomposition with fewer terms. Various approximation methods will be relevant here.

Innovation

Use the above primitive + other methods to build quantum algorithms for various problems, including physics simulation and linear algebra problems, of complexity $\mathrm{polylog}(1/\epsilon)$.

Preliminary studies

- Recently[2] we showed a way to simulate the evolution operator, $A \propto e^{iH}$, using $O(\log(1/\epsilon)/\log\log(1/\epsilon))$ queries.
- Using the Fourier transform, we can decompose e^{-H} in terms of unitaries. This may lead to an algorithm of complexity $\mathrm{polylog}(1/\epsilon)$ for physics simulation.
- There are many decompositions of $A = 1/H$ in terms of unitaries. Can we exploit them to have an algorithm of complexity $\mathrm{polylog}(1/\epsilon)$ for applying inverses?
- Other linear algebra problems?

[2] D. Berry, A.M. Childs, R. Cleve, R. Kothari, and RS., arXiv:1312.1414 (STOC) and arXiv:1412.4687 (PRL).

Innovation

Use the above primitive + other methods to build quantum algorithms for various problems, including physics simulation and linear algebra problems, of complexity $\mathrm{polylog}(1/\epsilon)$.

Preliminary studies

- Recently[2] we showed a way to simulate the evolution operator, $A \propto e^{iH}$, using $O(\log(1/\epsilon)/\log\log(1/\epsilon))$ queries.
- Using the Fourier transform, we can decompose e^{-H} in terms of unitaries. This may lead to an algorithm of complexity $\mathrm{polylog}(1/\epsilon)$ for physics simulation.
- There are many decompositions of $A = 1/H$ in terms of unitaries. Can we exploit them to have an algorithm of complexity $\mathrm{polylog}(1/\epsilon)$ for applying inverses?
- Other linear algebra problems?

[2] D. Berry, A.M. Childs, R. Cleve, R. Kothari, and RS., arXiv:1312.1414 (STOC) and arXiv:1412.4687 (PRL).

Innovation

Use the above primitive + other methods to build quantum algorithms for various problems, including physics simulation and linear algebra problems, of complexity $\mathrm{polylog}(1/\epsilon)$.

Preliminary studies

- Recently[2] we showed a way to simulate the evolution operator, $A \propto e^{iH}$, using $O(\log(1/\epsilon)/\log\log(1/\epsilon))$ queries.
- Using the Fourier transform, we can decompose e^{-H} in terms of unitaries. This may lead to an algorithm of complexity $\mathrm{polylog}(1/\epsilon)$ for physics simulation.
- There are many decompositions of $A = 1/H$ in terms of unitaries. Can we exploit them to have an algorithm of complexity $\mathrm{polylog}(1/\epsilon)$ for applying inverses?
- Other linear algebra problems?

[2] D. Berry, A.M. Childs, R. Cleve, R. Kothari, and RS., arXiv:1312.1414 (STOC) and arXiv:1412.4687 (PRL).

Innovation

Use the above primitive + other methods to build quantum algorithms for various problems, including physics simulation and linear algebra problems, of complexity $\mathrm{polylog}(1/\epsilon)$.

Preliminary studies

- Recently[2] we showed a way to simulate the evolution operator, $A \propto e^{iH}$, using $O(\log(1/\epsilon)/\log\log(1/\epsilon))$ queries.
- Using the Fourier transform, we can decompose e^{-H} in terms of unitaries. This may lead to an algorithm of complexity $\mathrm{polylog}(1/\epsilon)$ for physics simulation.
- There are many decompositions of $A = 1/H$ in terms of unitaries. Can we exploit them to have an algorithm of complexity $\mathrm{polylog}(1/\epsilon)$ for applying inverses?
- Other linear algebra problems?

[2] D. Berry, A.M. Childs, R. Cleve, R. Kothari, and RS., arXiv:1312.1414 (STOC) and arXiv:1412.4687 (PRL).

Innovation

Use the above primitive + other methods to build quantum algorithms for various problems, including physics simulation and linear algebra problems, of complexity $\mathrm{polylog}(1/\epsilon)$.

Preliminary studies

- Recently[2] we showed a way to simulate the evolution operator, $A \propto e^{iH}$, using $O(\log(1/\epsilon)/\log\log(1/\epsilon))$ queries.
- Using the Fourier transform, we can decompose e^{-H} in terms of unitaries. This may lead to an algorithm of complexity $\mathrm{polylog}(1/\epsilon)$ for physics simulation.
- There are many decompositions of $A = 1/H$ in terms of unitaries. Can we exploit them to have an algorithm of complexity $\mathrm{polylog}(1/\epsilon)$ for applying inverses?
- Other linear algebra problems?

[2] D. Berry, A.M. Childs, R. Cleve, R. Kothari, and RS., arXiv:1312.1414 (STOC) and arXiv:1412.4687 (PRL).

Opportunities and challenges

Surely, applying operations such as $1/H$ efficiently is important. But...

- In practice, linear systems of equations are not solved by multiplying a vector by $1/H$. A lot of computations can be reused to solve following instances ($H = LU$).
- Conditioning numbers can be very large
- Reading the full answer or encoding the initial vector can still be time consuming

Challenge: Do we have killer apps? (Stephen's talk)

We need more progress

Opportunities and challenges

Surely, applying operations such as $1/H$ efficiently is important. But...

- In practice, linear systems of equations are not solved by multiplying a vector by $1/H$. A lot of computations can be reused to solve following instances ($H = LU$).
- Conditioning numbers can be very large
- Reading the full answer or encoding the initial vector can still be time consuming

Challenge: Do we have killer apps? (Stephen's talk)

We need more progress

Opportunities and challenges

Surely, applying operations such as $1/H$ efficiently is important. But...

- In practice, linear systems of equations are not solved by multiplying a vector by $1/H$. A lot of computations can be reused to solve following instances ($H = LU$).
- Conditioning numbers can be very large
- Reading the full answer or encoding the initial vector can still be time consuming

Challenge: Do we have killer apps? (Stephen's talk)

We need more progress

Future directions

- Keep reducing the complexity of the algorithms
- Generalize known tools, and introduce new ones, for other applied math problems (e.g., solving differential equations)
- Find important applications (physics simulation, heat equations, ...)

Future directions

- Keep reducing the complexity of the algorithms
- Generalize known tools, and introduce new ones, for other applied math problems (e.g., solving differential equations)
- Find important applications (physics simulation, heat equations, ...)

Future directions

- Keep reducing the complexity of the algorithms
- Generalize known tools, and introduce new ones, for other applied math problems (e.g., solving differential equations)
- Find important applications (physics simulation, heat equations, ...)

DOE support

“The mission of the Advanced Scientific Computing Research (ASCR) program is to discover, develop, and deploy computational and networking capabilities to analyze, model, simulate, and predict complex phenomena important to the Department of Energy (DOE). A particular challenge of this program is fulfilling the science potential of emerging computing systems and other **novel computing architectures**, which will require numerous significant modifications to today’s tools and techniques to deliver on the promise of exascale science."

- Materials science
- Physics
- Chemistry

DOE support

“The mission of the Advanced Scientific Computing Research (ASCR) program is to discover, develop, and deploy computational and networking capabilities to analyze, model, simulate, and predict complex phenomena important to the Department of Energy (DOE). A particular challenge of this program is fulfilling the science potential of emerging computing systems and other **novel computing architectures**, which will require numerous significant modifications to today’s tools and techniques to deliver on the promise of exascale science."

- Materials science
- Physics
- Chemistry

Thank you.