Quantum Sensing and Information Processing

Lecture 6: Quantum Computing Algorithms – Part 2

Andreas Baertschi
Los Alamos National Laboratory

August 1st, 2019

LLNL-PRES-xxxxxx
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

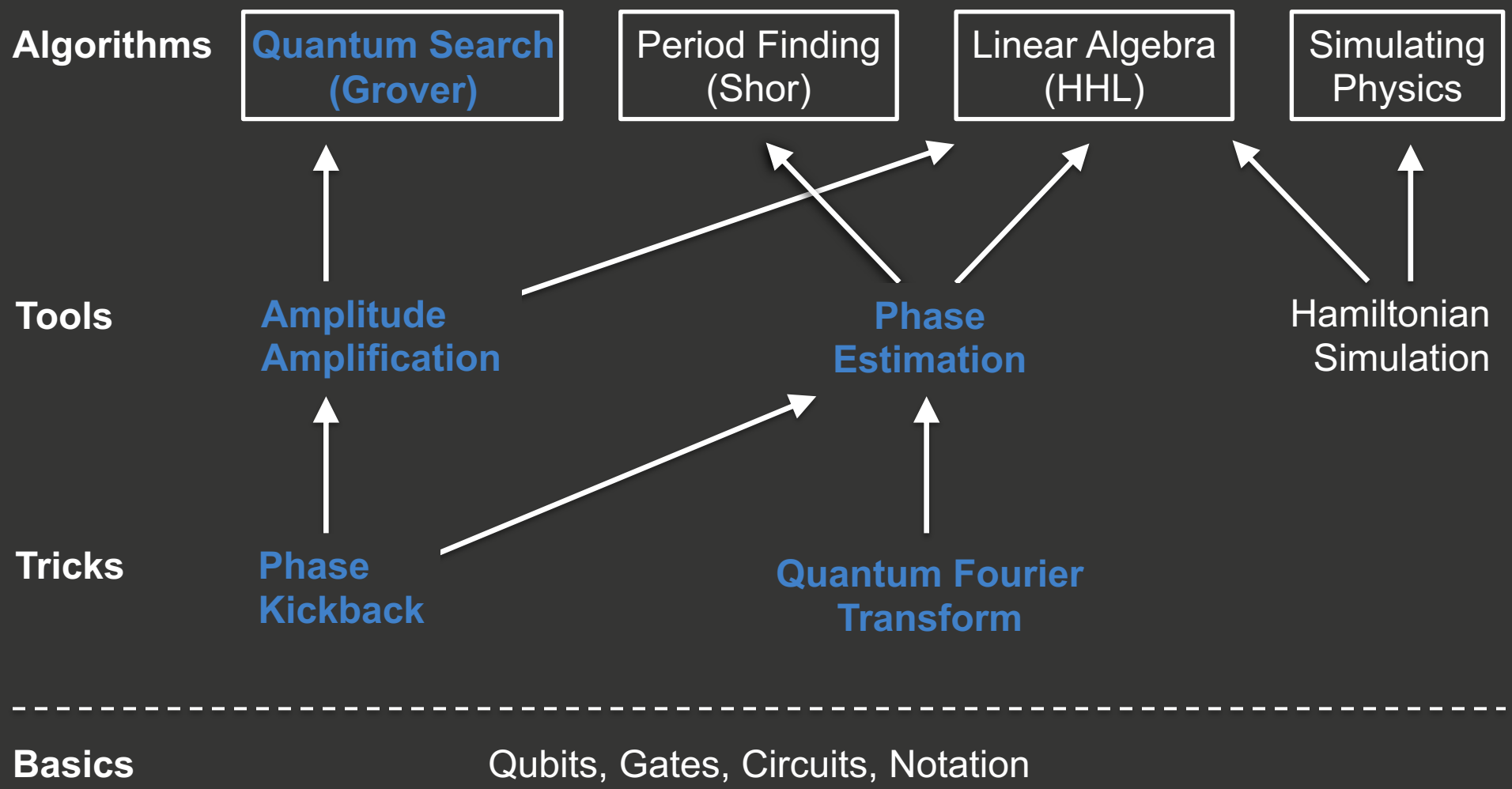
Quantum Computing Algorithms
Algorithms

LLNL CASIS Quantum Sensing and Information Processing Series

Andreas Bärtschi
baertschi@lanl.gov

July 31 / August 1, 2019

Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA


Algorithms
Quantum Search (Grover)
Period Finding (Shor)
Linear Algebra (HHL)
Simulating Physics
Tools
Amplitude Amplification
Phase Estimation
Hamiltonian Simulation
Tricks
Phase Kickback
Quantum Fourier Transform
Basics
Qubits, Gates, Circuits, Notation

Theory (Slides) and Practice (Quirk)

https://cnls.lanl.gov/~baertschi/QCA/

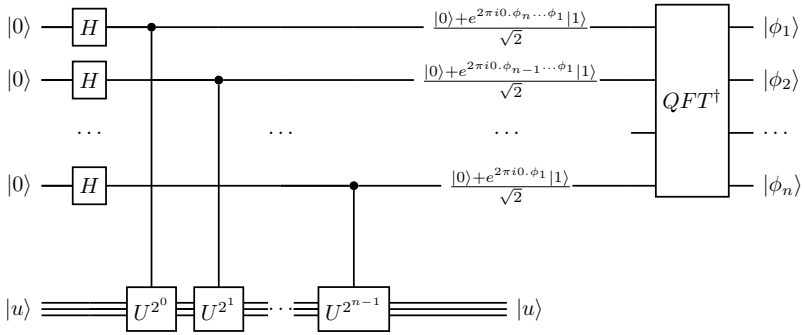
Recap

Amplitude Amplification

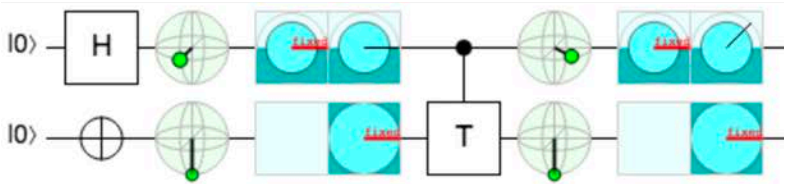
Given algorithm A with success probability *p*, one can do

- **Classical**
  Repeating A for 1/*p* times
- **Quantum**
  Boost amplitudes over √1/p rounds

Phase Estimation



Phase Kickback



Quantum Fourier Transform

QFT

$$\bigotimes_{k=n}^{1} |\phi_k\rangle \quad \bigotimes_{k=1}^{n} \left(|0\rangle + e^{2\pi i\, 0.\phi_k \ldots \phi_1}\, |1\rangle\right)$$

Inverse QFT

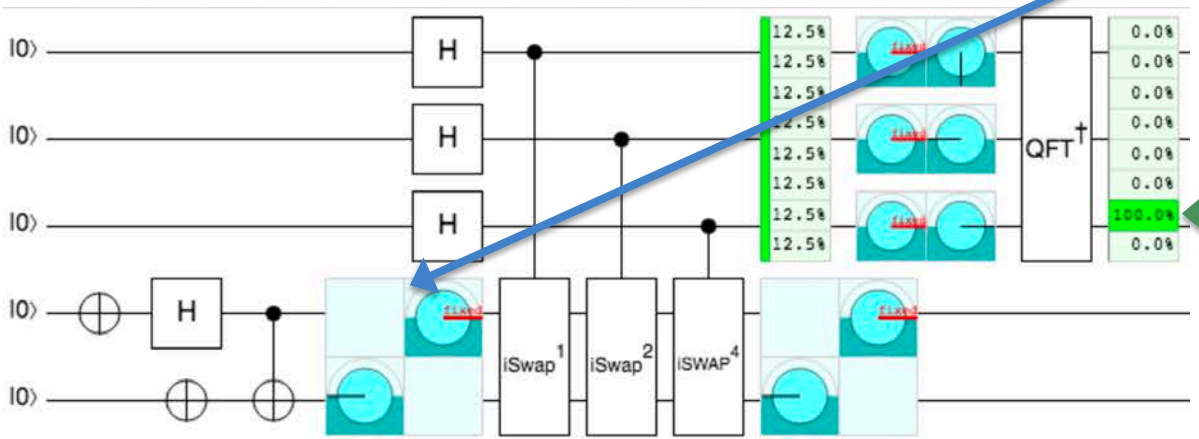
Recap Phase Estimation

Estimating Phases for the iSWAP gate:

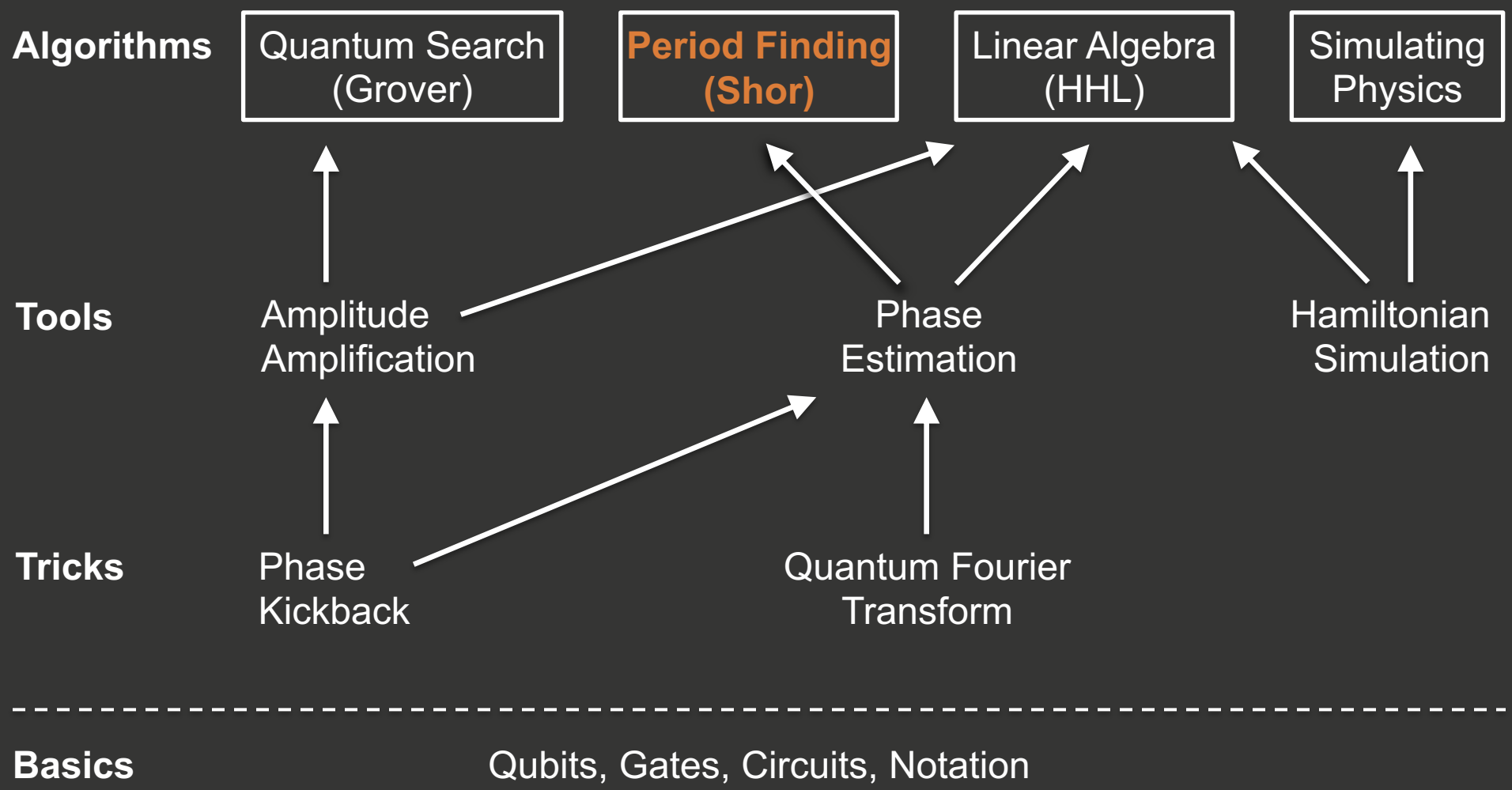
$$\begin{array}{cc} & \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{array}{c} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \end{array}$$

With Eigenstates and corresponding Eigenvalues:

$\|00\rangle$	$\frac{\|01\rangle + \|10\rangle}{\sqrt{2}}$	$\frac{\|01\rangle - \|10\rangle}{\sqrt{2}}$	$\|11\rangle$
1	i	$-i$	1



$e^{2\pi i\, 0.110}$


Algorithms
Quantum Search (Grover)
Period Finding (Shor)
Linear Algebra (HHL)
Simulating Physics
Tools
Amplitude Amplification
Phase Estimation
Hamiltonian Simulation
Tricks
Phase Kickback
Quantum Fourier Transform
Basics
Qubits, Gates, Circuits, Notation

Integer Factorization (Shor)

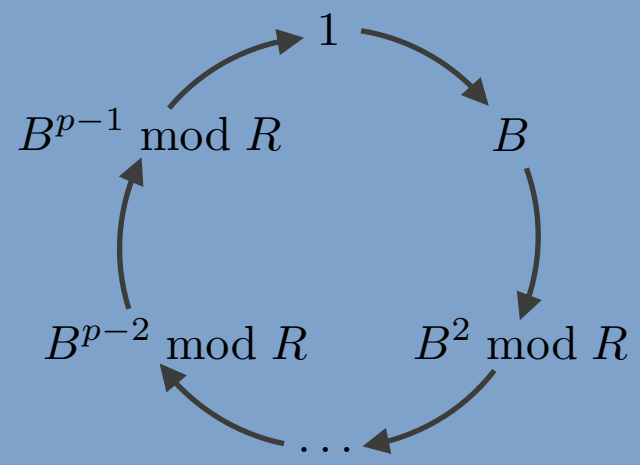
Given *n*-bit (composite) integer *R*, find a proper factor *f*

Classical Part:
Pre- / Post-Processing, Loop

1. Discard "simple" cases:
   R even, *R* prime power, (*R* prime)
2. Randomly choose 0 < *B* < *R*
   - check $f = \gcd(B, R)$
3. Compute period *p* of *B*
4. If *p* is even:
   - check $f = \gcd(B^{p/2} + 1, R)$
5. Repeat if necessary

Quantum Part:
Find Period *p* of

$$B^x \equiv 1 \pmod{R}$$



Period Finding I (Shor)

If *B* has period *p* modulo *R*, the unitary operator $\times B \pmod{R}$ has:

- *p* Eigenstates

$$|u_s\rangle = \frac{1}{\sqrt{p}} \sum_{k=0}^{p-1} e^{-2\pi i k s/p} \left| B^k \bmod R \right\rangle$$

- with Eigenvalues $e^{2\pi i s/p}$ Phase Estimation!

Need phase s/p to determine the period *p*,

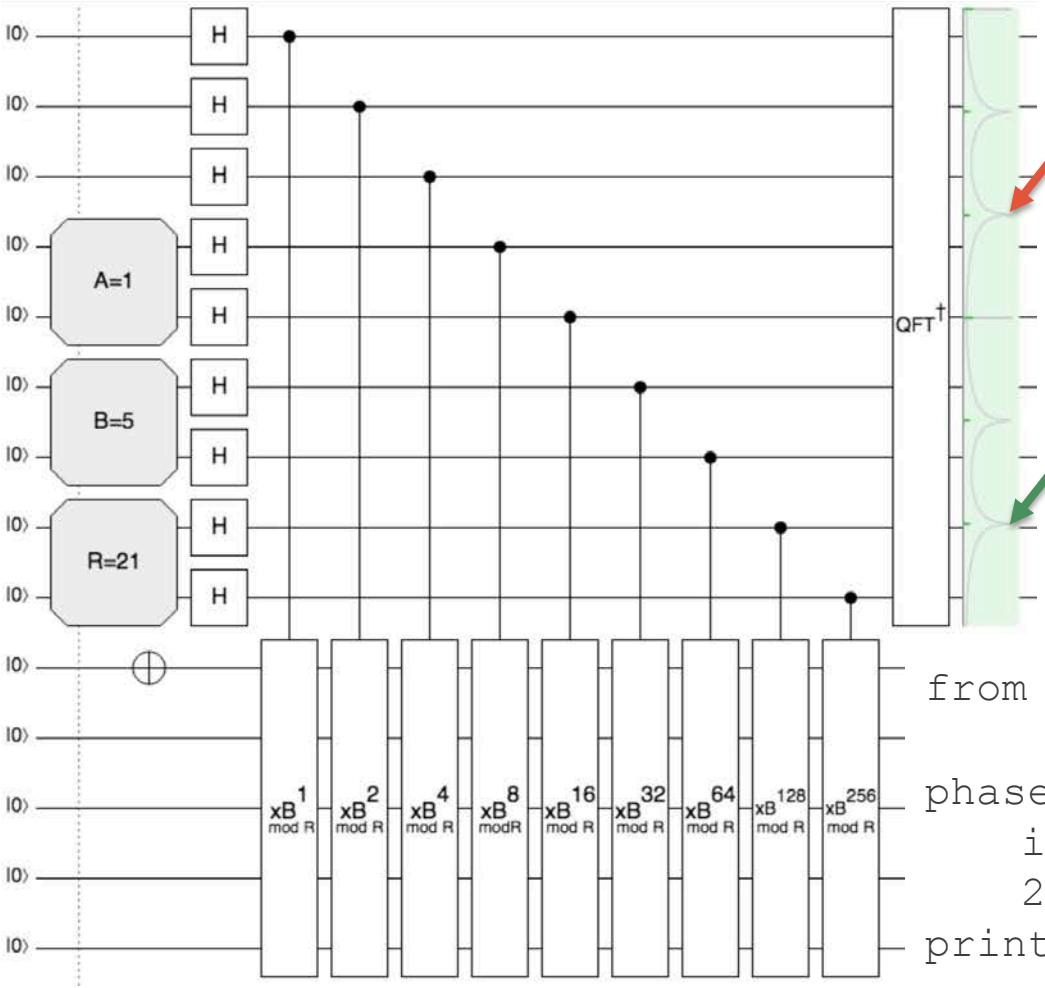
but need the period *p* to create $|u_s\rangle$.

Start in $|1\rangle = \frac{1}{\sqrt{p}} \sum_{s=0}^{p-1} |u_s\rangle$ **!!**

$5^k \bmod 21:$

$$|5\rangle \to |4\rangle \to |20\rangle \to |16\rangle \to |17\rangle \to |1\rangle$$

Period Finding II (Shor)



Phase $s/p \approx 2/6 = 1/3$
gives incorrect period
⇒ fail ⇒ repeat

Phase $427/512 \approx 5/6$

Use continued fractions
to find the “real fraction”

```
from fractions import Fraction

phase = Fraction(
    int(“110101011”, 2),
    2**9)
print(phase.limit_denominator(21))
```

Extra square roots (Shor)

If *R* is composite and not a prime power,
the number 1 has at least four square roots modulo *R* (not only 1, -1):

$$B^p \equiv 1 \pmod{R}$$
$$B^{p/2} \cdot B^{p/2} \equiv 1 \pmod{R}$$

$$(B^{p/2} - 1) \cdot (B^{p/2} + 1) \equiv 0 \pmod{R}$$

Want / hope for: $B^{p/2} + 1 \not\equiv 0 \pmod{R}$

Integer Factorization (Shor)

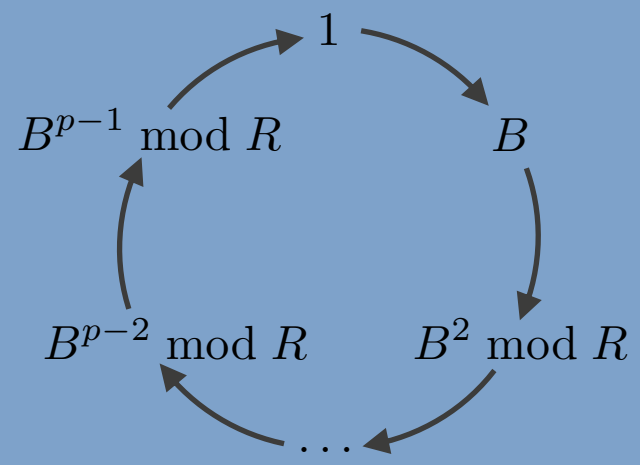
Given *n*-bit integer *R*, find a proper factor *f*

Classical Part:
Pre- / Post-Processing, Loop

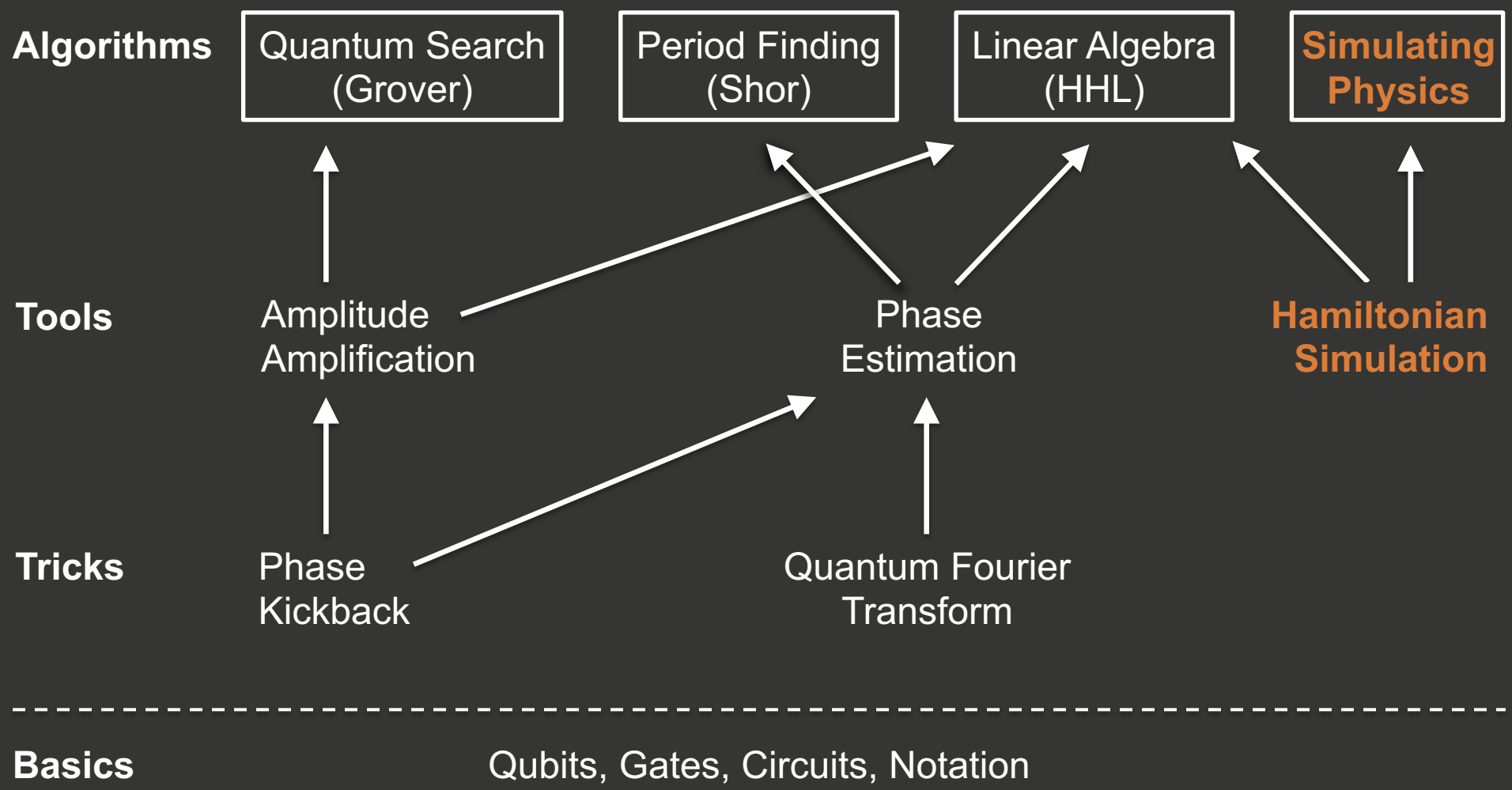
1. Discard "simple" cases:
   R even, *R* prime, *R* prime power
2. Randomly choose 0 < *B* < *R*
   - check $f = \gcd(B, R)$
3. Compute period *p* of *B*
4. If *p* is even:
   - check $f = \gcd(B^{p/2} + 1, R)$
5. Repeat if necessary (~ prob. 0.75)

Quantum Part:
Find Period *p* of

$$B^x \equiv 1 \pmod{R}$$



Everything $poly(n)$ vs. classical $e^{\sqrt[3]{n}}$: **Quantum Speedup: Superpolynomial**


Algorithms
Quantum Search (Grover)
Period Finding (Shor)
Linear Algebra (HHL)
Simulating Physics
Tools
Amplitude Amplification
Phase Estimation
Hamiltonian Simulation
Tricks
Phase Kickback
Quantum Fourier Transform
Basics
Qubits, Gates, Circuits, Notation

Hamiltonian Simulation
Problem Definition

Schrödinger Equation:

$$i\frac{d\,|\Psi(t)\rangle}{dt} = H\,|\Psi(t)\rangle$$

Time-independent Hamiltonian:

$$|\Psi(t)\rangle = e^{-itH}\,|\Psi(0)\rangle$$

Task:

Find a quantum circuit, that simulates e^{-itH} as close as possible!

Hamiltonian Simulation

Two notable Problem Types

From Physics

- *H* is a sum of Pauli Tensor Product terms:

$$H = \sum_{j=1}^{m} H_j$$

- with, e.g., acting on qubits 3, 2, 1, respectively.

$$H_j = Z_3 \otimes Y_2 \otimes X_1$$

- Structure from Physics through Jordan-Wigner transform.

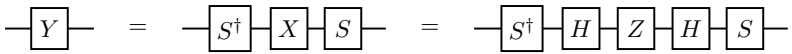
From Maths & CS

- *H* is a *s*-sparse (at most *s* non-zero entries per row and per column)
- Given row *i* and number *x*, there must be a fast way ("Oracle") to get
  - the index *j* and
  - the matrix entry $H_{i,j}$

  of xth non-zero entry in row *i.*

Pauli Tensor Products
can be simulated exactly by **Diagonalization**

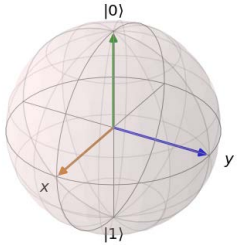
Axis Change



$$Y = S \cdot H \cdot Z \cdot H \cdot S^\dagger$$

$$\Rightarrow e^{-itY} = e^{-it \cdot S \cdot H \cdot Z \cdot H \cdot S^\dagger}$$

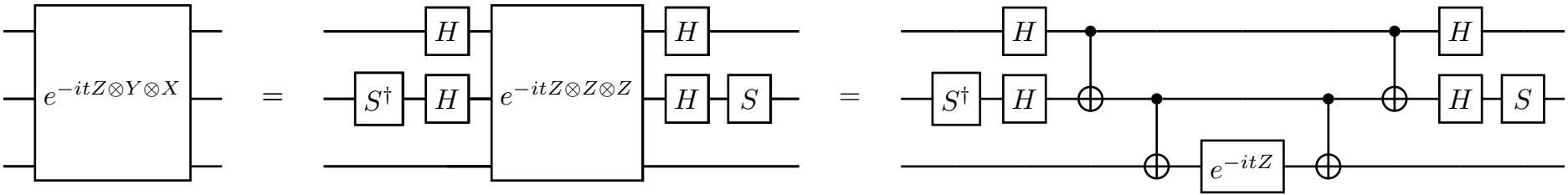
$$= S \cdot H \cdot e^{-it \cdot Z} \cdot H \cdot S^\dagger$$



Parity

$$Z \otimes Z \otimes Z = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

Diagonalization & Break down



Sum of (non-commuting) H_j

$$H = \sum_j H_j, \quad \text{e.g.} \quad H = X + Z$$

Problem

If the H_j do not commute,

$$X \cdot Z = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = Z \cdot X$$

then the exponential does not factor:

$$e^{-it\cdot(X+Z)} \neq e^{-itX} \cdot e^{-itZ}$$

Suzuki-Trotter

Small alternating H_j steps

$$H = \sum_{j=1}^{m} H_j = X + Z$$

Approximate by:

$$U_1(t,r) = \left(\prod_{j=1}^{m} e^{-i\frac{t}{r}H_j} \right)^r$$

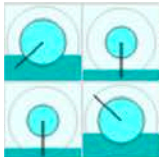
$$U_2(t,r) = \left(\prod_{j=1}^{m} e^{-i\frac{t}{2r}H_j} \prod_{j=m}^{1} e^{-i\frac{t}{2r}H_j} \right)^r$$

$$U_{2k}(t,r) = \left([U_{2k-2}(tp_k,r)]^2 U_{2k-2}((1-4p_k),r)[U_{2k-2}(tp_k,r)]^2 \right)^r$$

$$\text{with } p_k = 1/(4 - 4^{1/(2k-1)})$$

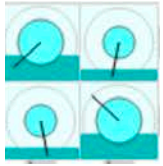
Example, $t = \pi/2$:

- Exact: numpy



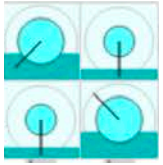
$e^{-i\pi/2(X+Z)}$

- Trotter: 16 gates



$U_1(\pi/2, 8)$

- Suzuki: 9 gates



$U_2(\pi/2, 4)$

Hamiltonian Simulation Results

Number of "easy" exponentials for error $\leq \varepsilon$ and time *t*

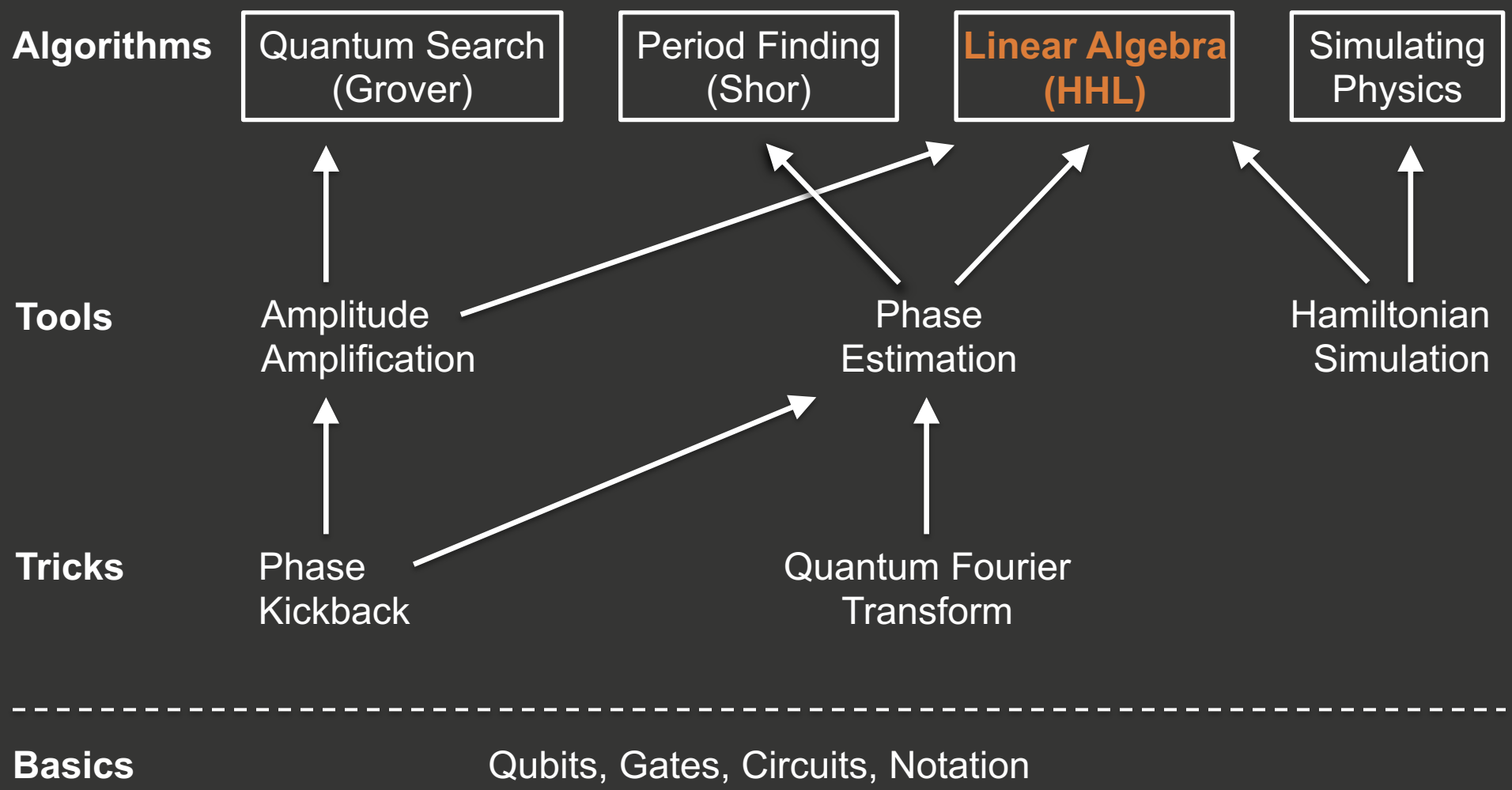
- From Physics: Sum of Pauli Tensor Product terms

$$N_{exp} \leq 5^{2k} 2m (m \cdot t \cdot ||H||)^{1+1/2k} / \varepsilon^{1/2k} \approx \widetilde{\mathcal{O}}(m^2 \cdot t \cdot ||H||)$$

- From Mathematics:
  Decompose *s*-sparse *H* into Sum of $m = 6s^2$ 1-sparse Hamiltonians, then use techniques from above:

$$N_{exp} \approx \widetilde{\mathcal{O}}(s^4 \cdot t \cdot ||H||), \qquad \text{plus } \widetilde{\mathcal{O}}(\log N s^4 \cdot t \cdot ||H||) \text{ extra gates.}$$

Quantum Speedup: Exponential


Algorithms
Quantum Search (Grover)
Period Finding (Shor)
Linear Algebra (HHL)
Simulating Physics
Tools
Amplitude Amplification
Phase Estimation
Hamiltonian Simulation
Tricks
Phase Kickback
Quantum Fourier Transform
Basics
Qubits, Gates, Circuits, Notation

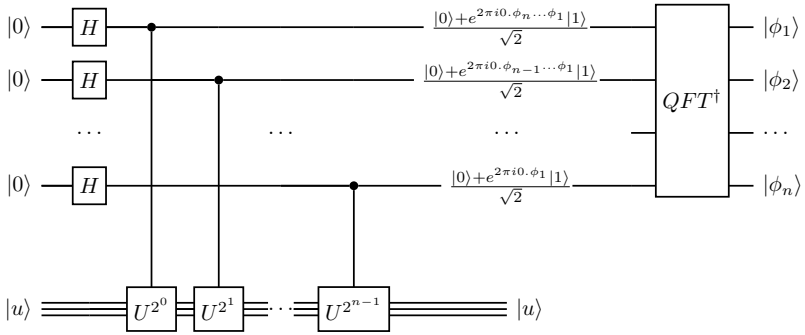
Recall Recap

Amplitude Amplification

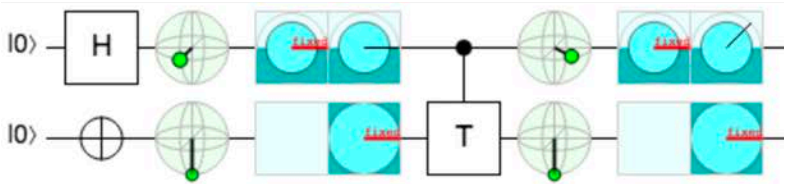
Given algorithm A with success probability *p*, one can do

- **Classical**
  Repeating A for 1/*p* times
- **Quantum**
  Boost amplitudes over √1/p rounds

Phase Estimation



Phase Kickback



Quantum Fourier Transform

QFT

$$\bigotimes_{k=n}^{1} |\phi_k\rangle \quad \bigotimes_{k=1}^{n} \left(|0\rangle + e^{2\pi i\, 0.\phi_k \ldots \phi_1}\, |1\rangle\right)$$

Inverse QFT

Linear System Problems

$$A \cdot \vec{x} = \vec{b}$$

Classical and Quantum (Harrow, Hassidim, Lloyd)

Classical problem

Conjugate gradient method: $\mathcal{O}(Ns\sqrt{\kappa}\log(1/\varepsilon))$

NxN Matrix *A*:

- Condition number κ
- Sparsity *s*
- *A* positive semidefinit

Solves $\vec{x} = A^{-1} \cdot \vec{b}$

Quantum problem

HHL: $\mathcal{O}(\kappa T_B + \log(N) T_A s^4 \kappa^2/\varepsilon)$

Additionally:

- Eigenvalues in [1/κ, 1]
- $|b\rangle = \vec{b}/||\vec{b}||$ preparable in time T_B
- Oracle access to *A* in time T_A
- *A* Hermitian, $A = A^\dagger$
- **no interest in all entries of $\vec{x}$**

Solves $|x\rangle = \dfrac{A^{-1}|b\rangle}{||A^{-1}|b\rangle||}$

Quantum Speedup: depends…

Quantum Linear System Problem (HHL)

Example

$$A \cdot \vec{x} = \vec{b}: \qquad \begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

$$\vec{x} = A^{-1} \cdot \vec{b} = \begin{pmatrix} 1.5 & -0.5 \\ -0.5 & 1.5 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \end{pmatrix}$$

$$|x\rangle = \frac{1}{\sqrt{10}} \begin{pmatrix} 3 \\ -1 \end{pmatrix}$$

Eigenstates	**Eigenvalue**
$\lvert+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	**1**
$\lvert-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	**0.5**

$$|b\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle = \frac{|+\rangle + |-\rangle}{\sqrt{2}}$$

Quantum Linear System Problem (HHL)

Helpful Facts

Let *A* have Eigenstates $|u_j\rangle$

with Eigenvalues $\lambda_j \in [1/\kappa, 1]$:

- We can write |b⟩ as
  $\sum \beta_j |u_j\rangle$
- We can write |x⟩ as
  $\approx \sum \frac{\beta_j}{\lambda_j} |u_j\rangle$

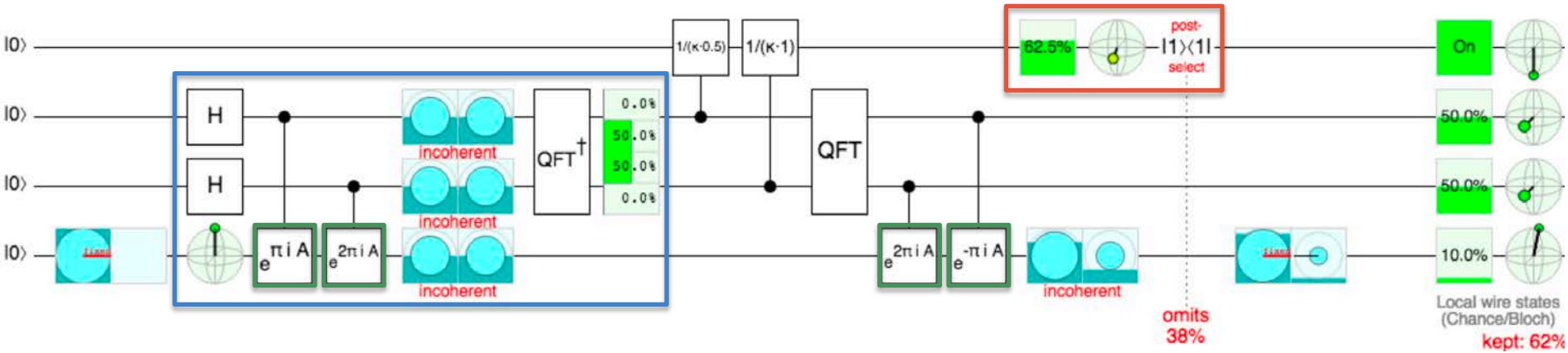
Need to find the Eigenvalues!

- $e^{it\cdot A}$ has Eigenvalues $e^{it\cdot\lambda_j}$
  in particular
  $$e^{i\pi A}|u_j\rangle = e^{i\pi\lambda_j}|u_j\rangle = e^{2\pi i\cdot\lambda_j/2}|u_j\rangle$$
  has a phase of
  $$\lambda_j/2 \in [1/(2\kappa), 1/2] \subset [0, 1)$$

Using Hamiltonian Simulation we can do Phase Estimation!

Quantum Linear System Problem (HHL)

Hamiltonian Simulation | Phase Estimation | Amplitude Amplification

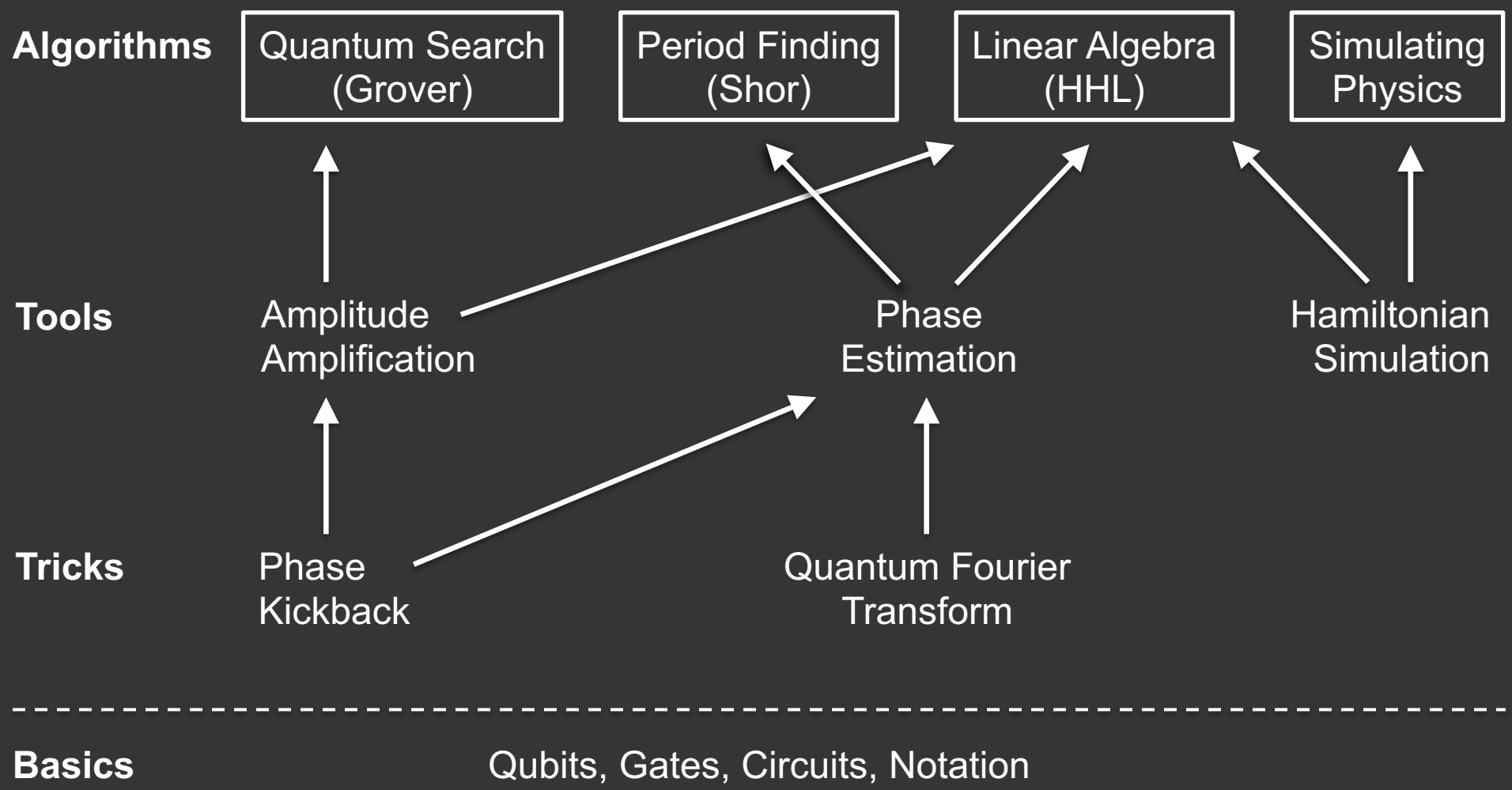


$$|b\rangle = \sum \beta_j |u_j\rangle |0\rangle_r |0\rangle_a \rightarrow \sum \beta_j |u_j\rangle |\lambda_j\rangle_r |0\rangle_a \qquad \text{Phase Estimation}$$

$$\rightarrow \sum \beta_j |u_j\rangle |\lambda_j\rangle_r \left(\sqrt{1-\frac{1}{\kappa^2\lambda_j^2}}\,|0\rangle + \frac{1}{\kappa\lambda_j}|1\rangle\right)_a \qquad \text{Conditional Rotation}$$

$$\rightarrow \sum \beta_j |u_j\rangle |0\rangle_r \left(\sqrt{1-\frac{1}{\kappa^2\lambda_j^2}}\,|0\rangle + \frac{1}{\kappa\lambda_j}|1\rangle\right)_a \qquad \text{uncompute}$$

$$\rightarrow \sqrt{\frac{\kappa^2}{\sum \beta_j^2/\lambda_j^2}} \sum \frac{\beta_j}{\lambda_j} |u_j\rangle |0\rangle_r |1\rangle_a \approx \sum \frac{\beta_j}{\lambda_j} |u_j\rangle \qquad \text{if we measured } |1\rangle_a$$


Algorithms
Quantum Search (Grover)
Period Finding (Shor)
Linear Algebra (HHL)
Simulating Physics
Tools
Amplitude Amplification
Phase Estimation
Hamiltonian Simulation
Tricks
Phase Kickback
Quantum Fourier Transform
Basics
Qubits, Gates, Circuits, Notation