Title: Classical Post-processing for Low-Depth Phase Estimation Circuits

Date: Jun 03, 2007 11:40 AM

URL: http://pirsa.org/07060018

Abstract: Traditionally, we use the quantum Fourier transform circuit (QFT) in order to perform quantum phase estimation, which has a number of useful applications. The QFT circuit for a binary field generally consists controlled-rotation gates which, when removed, yields the lower-depth approximate QFT circuit. It is known that a logarithmic-depth approximate QFT circuit is sufficient to perform phase estimation with a degree of accuracy negligibly lower than that of the full QFT. However, when the depth of the AQFT circuit becomes even lower, the phase estimation procedure no longer produces results that are immediately correlated to the desired phase. In this talk, I will explore the possibility of retrieving this information with classical analysis and with computer post-processing of the measured results of a low-depth AQFT circuit in a phase estimation algorithm.

Classical Post-processing for Low-Depth Phase Estimation Circuits

Donny Cheung

Department of Combinatorics & Optimization
Institute for Quantum Computing
University of Waterloo

June 3, 2007

Outline

1. Review of Phase Estimation and AQFT
   - Phase Estimation
   - Approximate Quantum Fourier Transform
2. Classical Post-processing
   - Maximum Likelihood Estimation
   - Classical Post-processing Algorithm
3. Hidden Subgroup Problems
   - Definitions
   - Dihedral HSP
   - Postprocessing for Dihedral HSP

Eigenvalue Estimation

- Given: unitary operator U and eigenstate $|u\rangle$
- Find the corresponding eigenvalue $\lambda = e^{2\pi ix}$
- We have $U|u\rangle = e^{2\pi ix}|u\rangle$
- We are given copies of an gate that performs controlled-U

Eigenvalue Estimation

- Consider x as a binary (base-2) fraction $x = 0.x_1x_2 \ldots x_n$
- Prepare n qubits in state $|0\rangle + e^{2\pi i(2^k x)}|1\rangle$ for $k = 0, 1, \ldots, n-1$
- $|0\rangle + e^{2\pi i(2^k x)}|1\rangle = |0\rangle + e^{2\pi i(0.x_{k+1}x_{k+2}\ldots)}|1\rangle$
- Apply inverse QFT to provide rotational corrections and measure x_{k+1} (Phase Estimation)

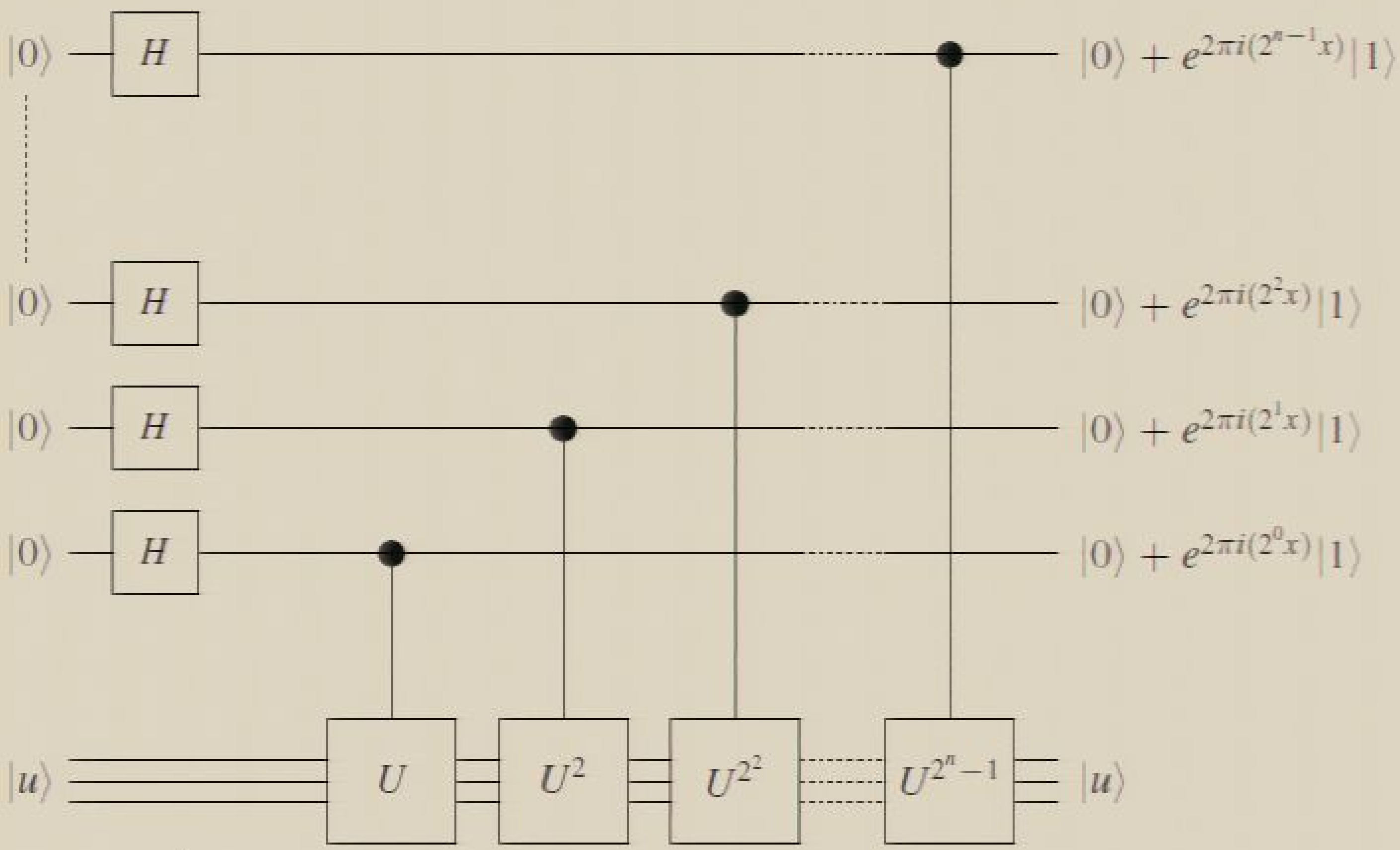
Eigenvalue Estimation

- Given: unitary operator U and eigenstate $|u\rangle$
- Find the corresponding eigenvalue $\lambda = e^{2\pi ix}$
- We have $U|u\rangle = e^{2\pi ix}|u\rangle$
- We are given copies of an gate that performs controlled-U

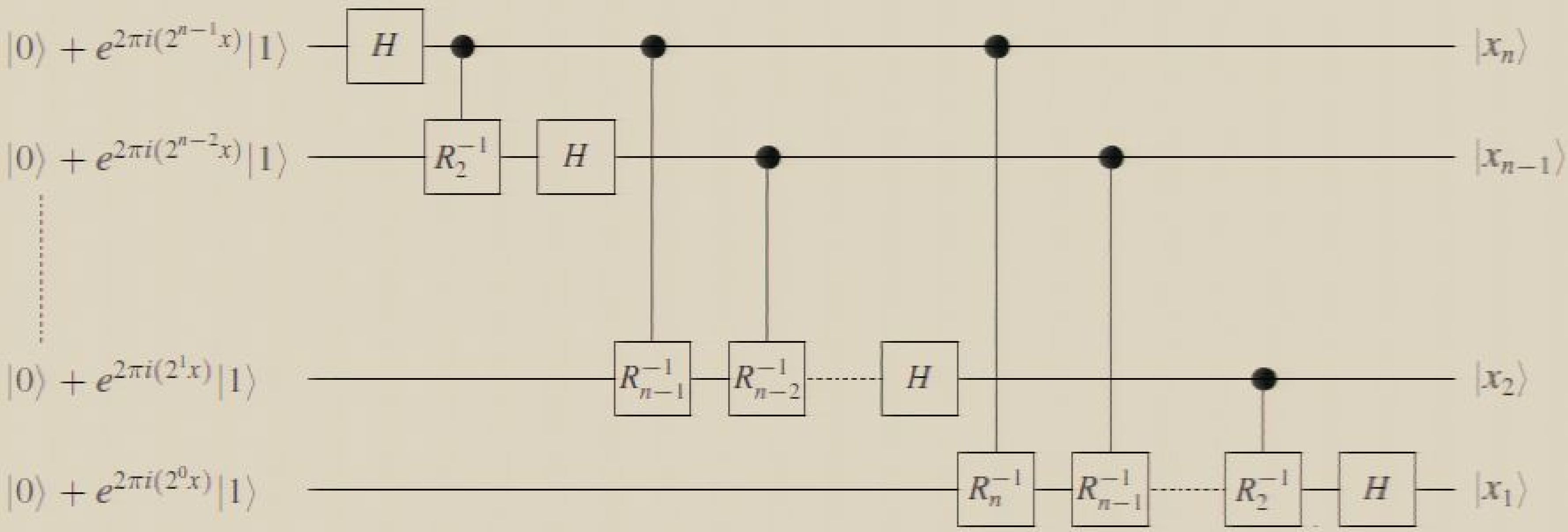
Eigenvalue Estimation

- Consider x as a binary (base-2) fraction $x = 0.x_1x_2 \ldots x_n$
- Prepare n qubits in state $|0\rangle + e^{2\pi i(2^k x)}|1\rangle$ for $k = 0, 1, \ldots, n-1$
- $|0\rangle + e^{2\pi i(2^k x)}|1\rangle = |0\rangle + e^{2\pi i(0.x_{k+1}x_{k+2}\ldots)}|1\rangle$
- Apply inverse QFT to provide rotational corrections and measure x_{k+1} (Phase Estimation)

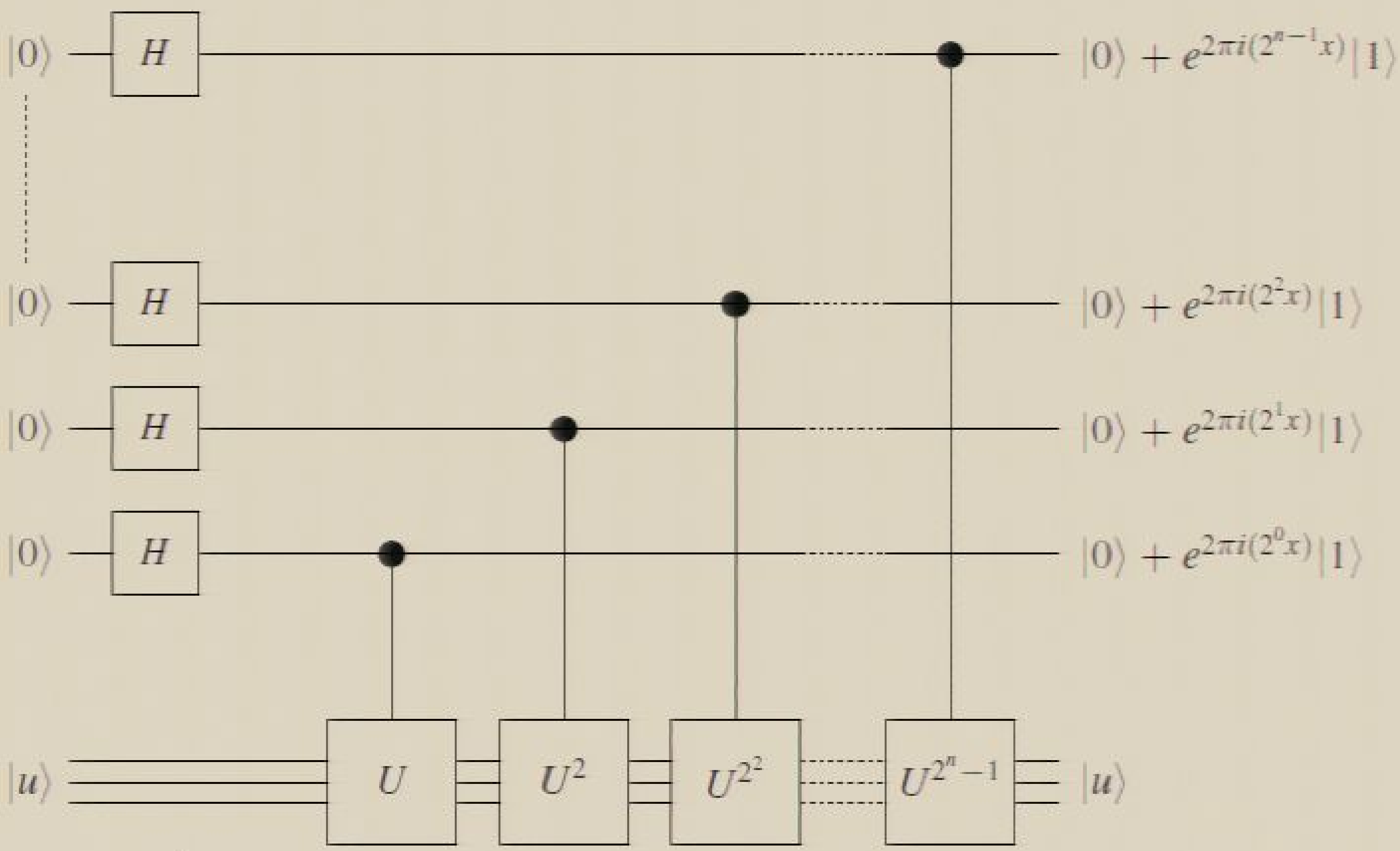
Eigenvalue Estimation Circuit



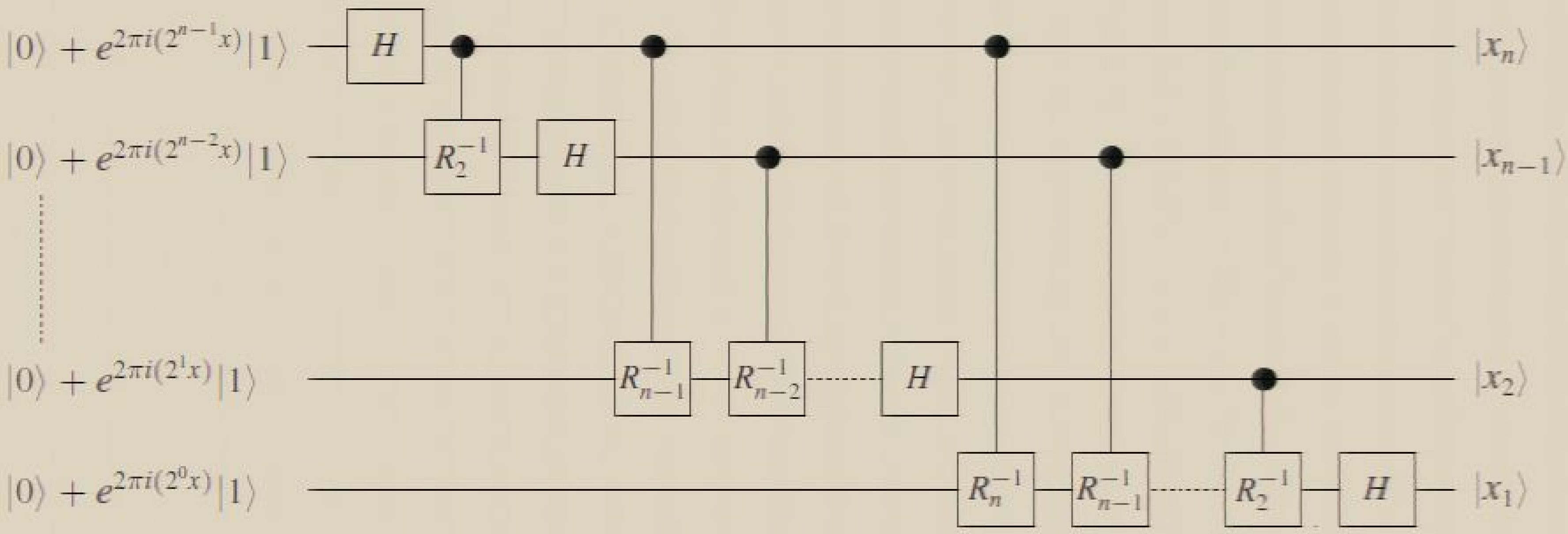
Phase Estimation with Inverse QFT



Eigenvalue Estimation Circuit



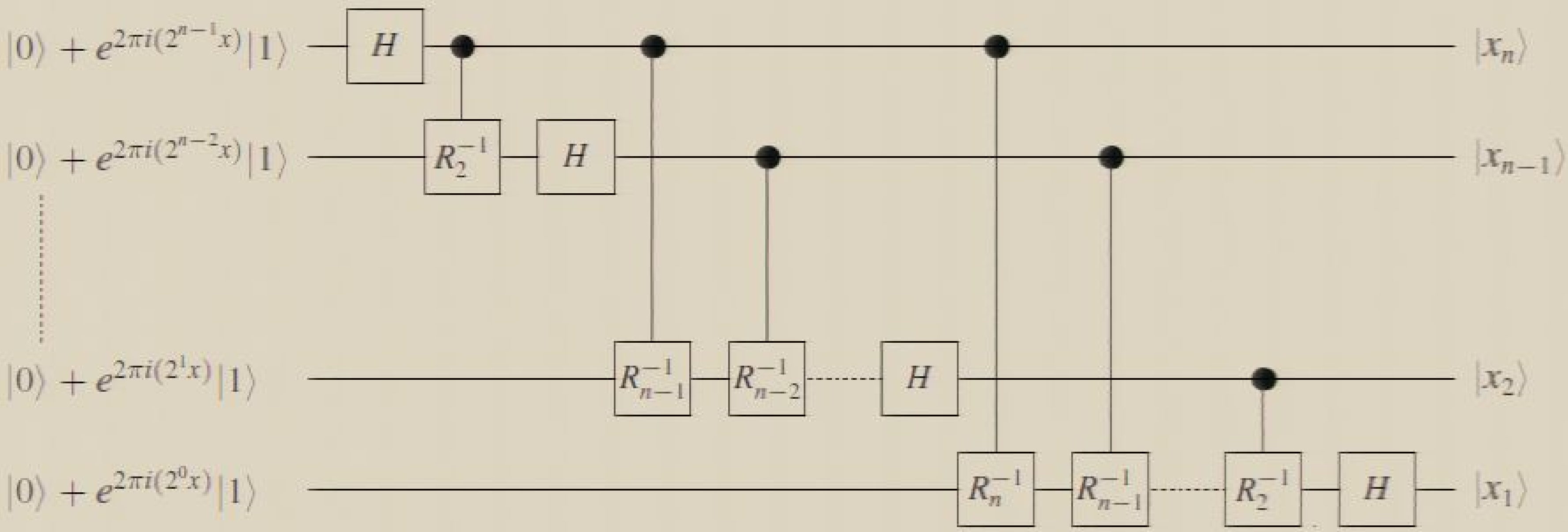
Phase Estimation with Inverse QFT



Semi-classical Phase Estimation

- In the Inverse QFT, qubits are used as quantum control, then measured
- We can replace this with measurement, followed by classical control
- Prepare the qubits of the output register one at a time
- Start with x_n and work down to x_1
- Use these results to classically control the phase rotation corrections

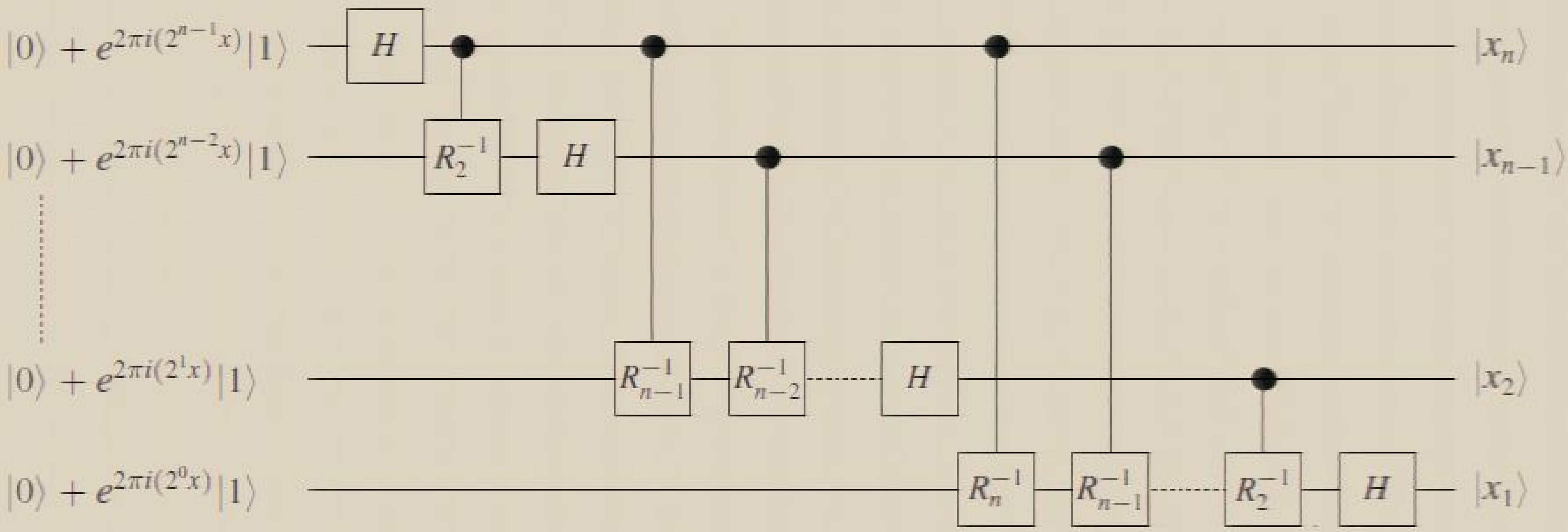
Phase Estimation with Inverse QFT



Semi-classical Phase Estimation

- In the Inverse QFT, qubits are used as quantum control, then measured
- We can replace this with measurement, followed by classical control
- Prepare the qubits of the output register one at a time
- Start with x_n and work down to x_1
- Use these results to classically control the phase rotation corrections

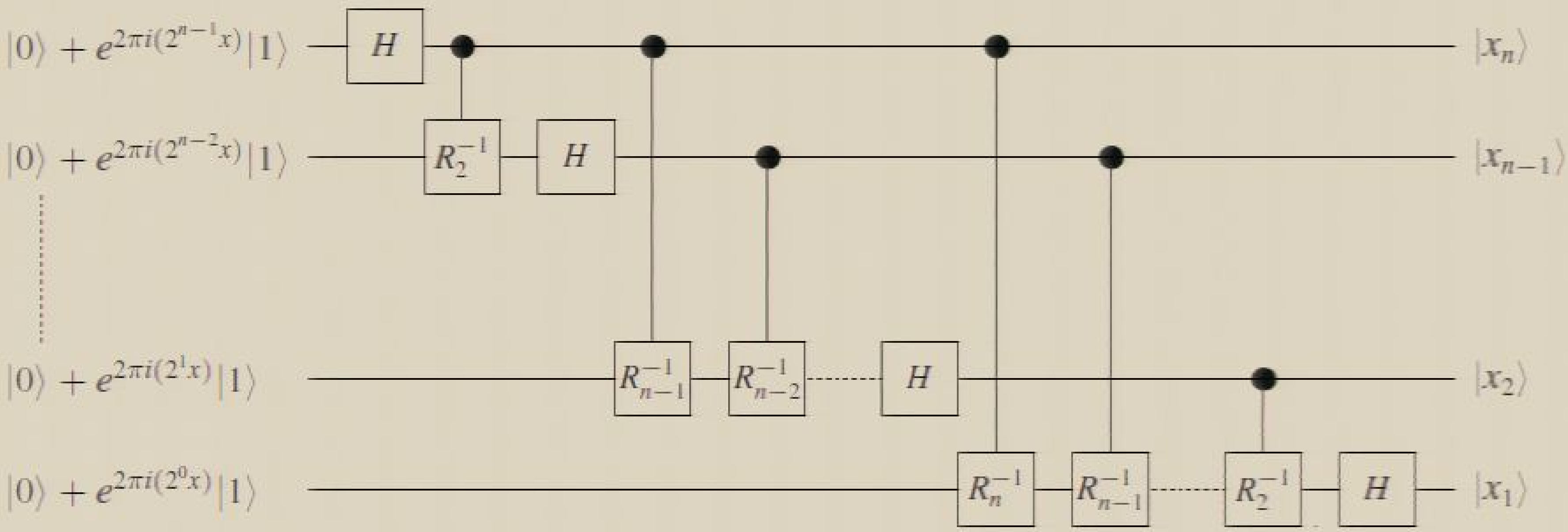
Phase Estimation with Inverse QFT



Semi-classical Phase Estimation

- In the Inverse QFT, qubits are used as quantum control, then measured
- We can replace this with measurement, followed by classical control
- Prepare the qubits of the output register one at a time
- Start with x_n and work down to x_1
- Use these results to classically control the phase rotation corrections

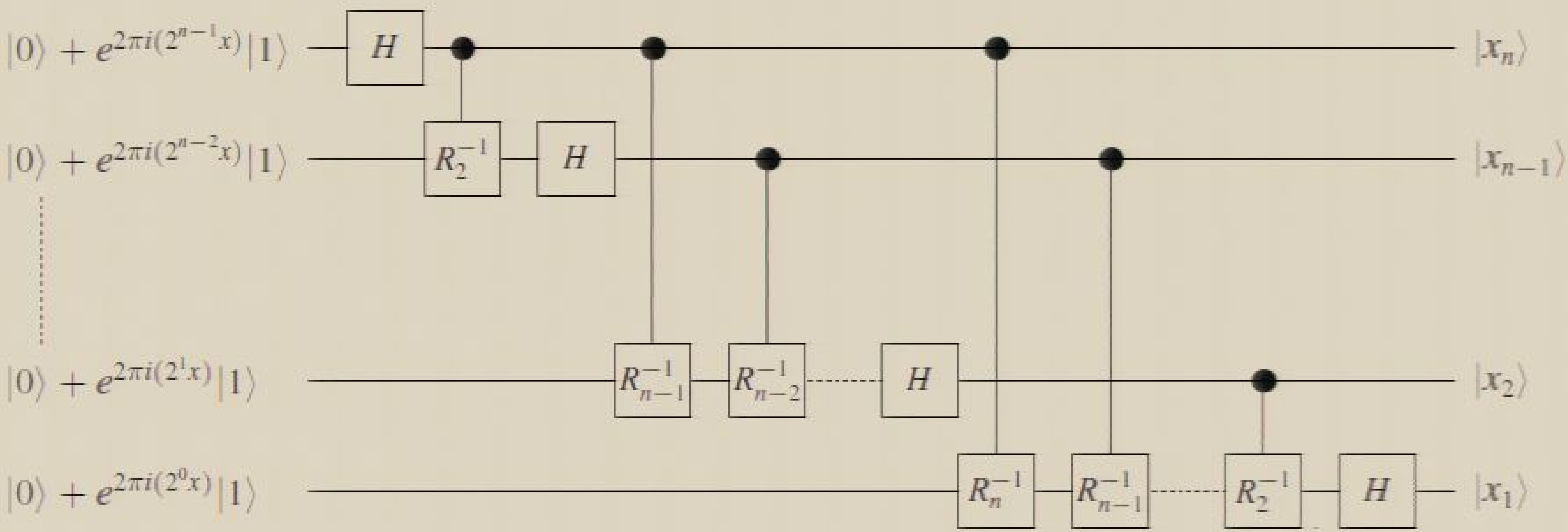
Phase Estimation with Inverse QFT



Semi-classical Phase Estimation

- In the Inverse QFT, qubits are used as quantum control, then measured
- We can replace this with measurement, followed by classical control
- Prepare the qubits of the output register one at a time
- Start with x_n and work down to x_1
- Use these results to classically control the phase rotation corrections

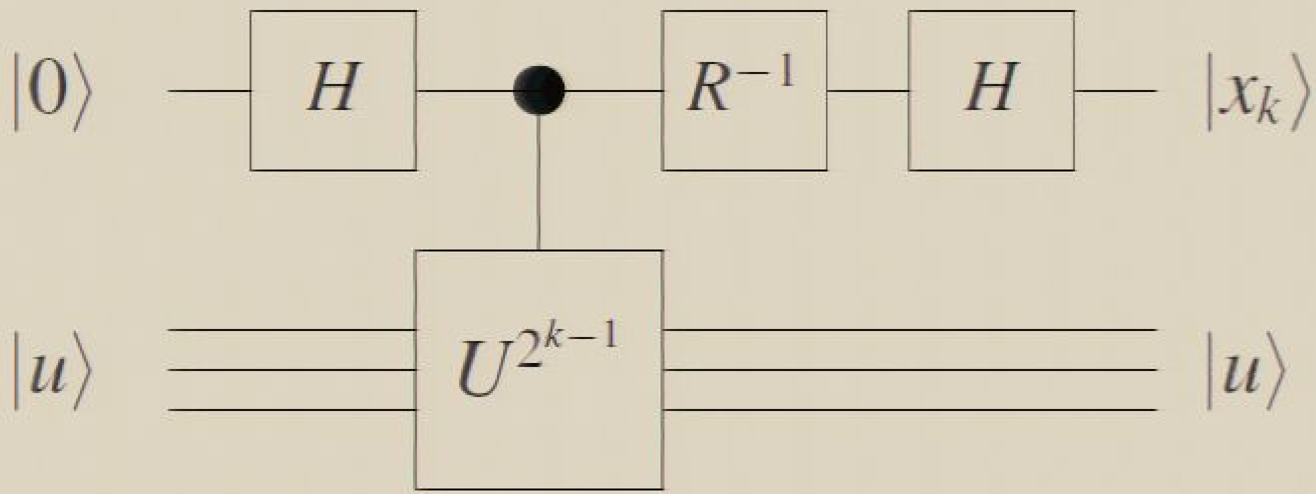
Phase Estimation with Inverse QFT



Semi-classical Phase Estimation

- In the Inverse QFT, qubits are used as quantum control, then measured
- We can replace this with measurement, followed by classical control
- Prepare the qubits of the output register one at a time
- Start with x_n and work down to x_1
- Use these results to classically control the phase rotation corrections

Circuit for sampling individual bit

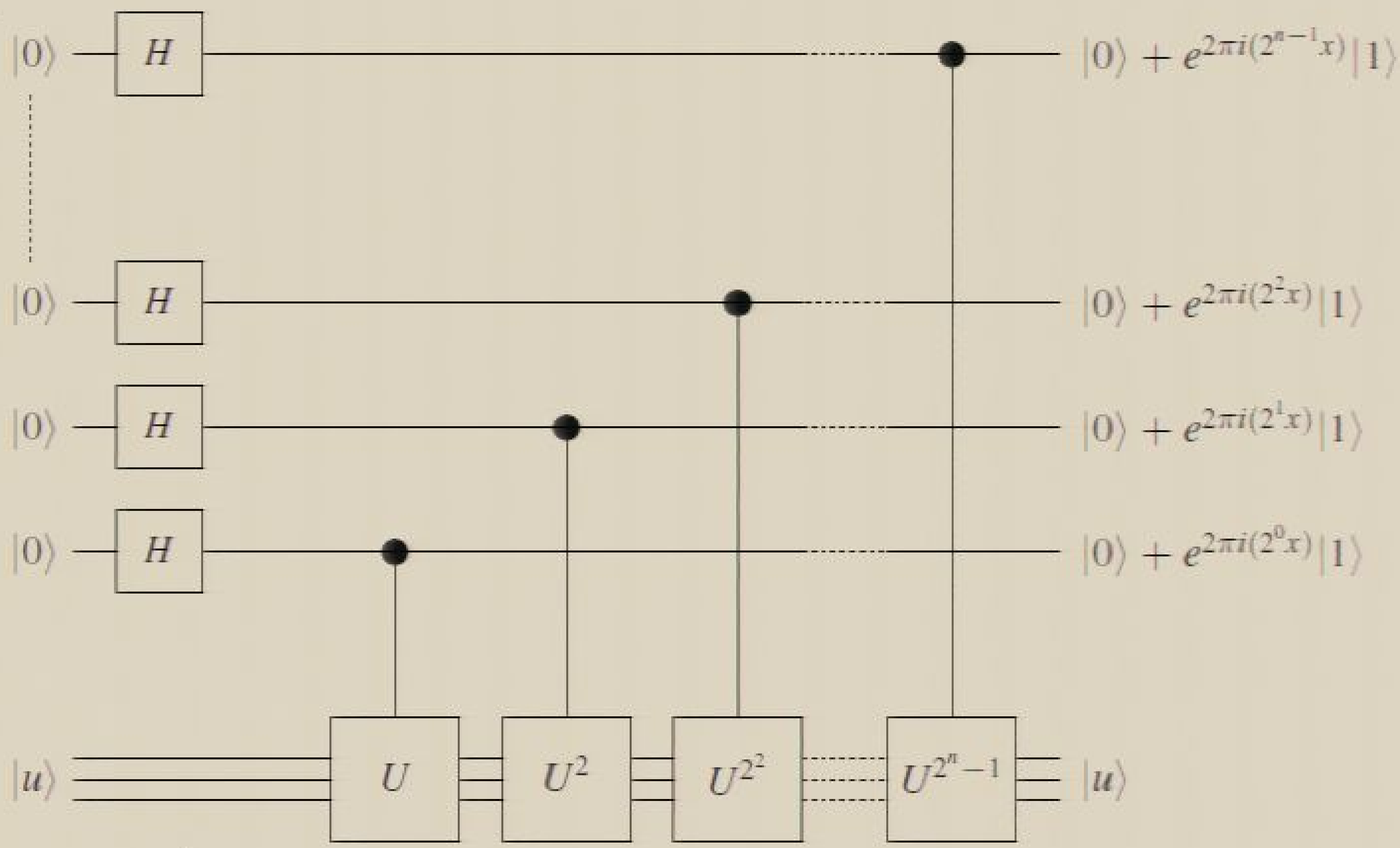


- In general, R estimates and tries to correct the phase shift $e^{2\pi i(2^{k-1}x-(0.x_k))}$
- We can improve the way we estimate R
- Example: multiple measurements for each bit x_k

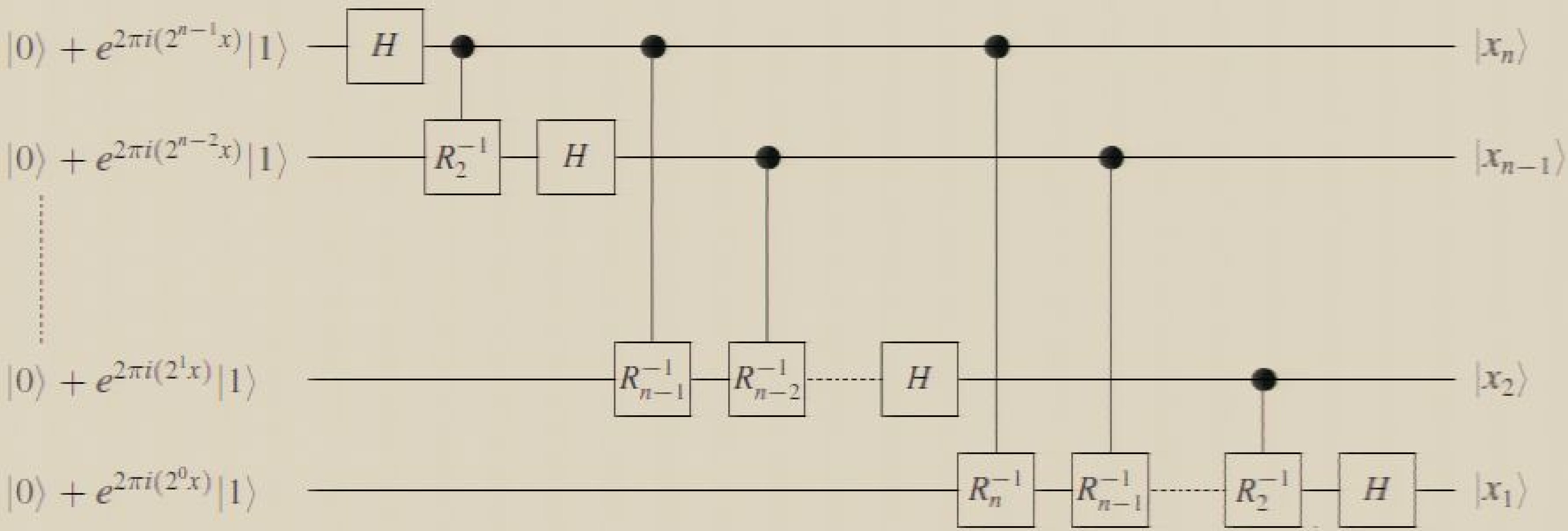
Semi-classical Phase Estimation

- In the Inverse QFT, qubits are used as quantum control, then measured
- We can replace this with measurement, followed by classical control
- Prepare the qubits of the output register one at a time
- Start with x_n and work down to x_1
- Use these results to classically control the phase rotation corrections

Eigenvalue Estimation Circuit



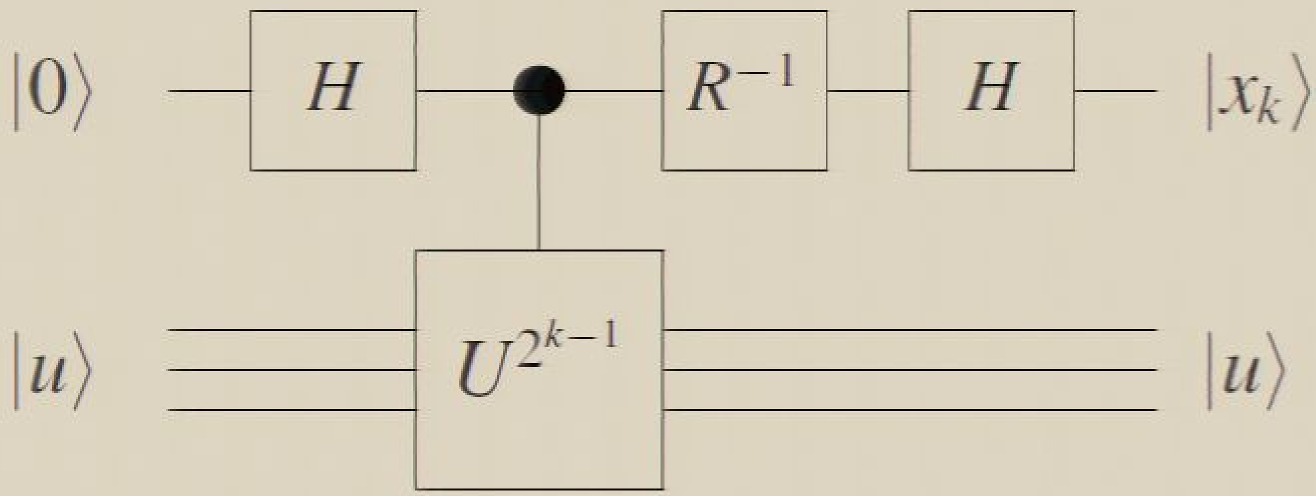
Phase Estimation with Inverse QFT



Semi-classical Phase Estimation

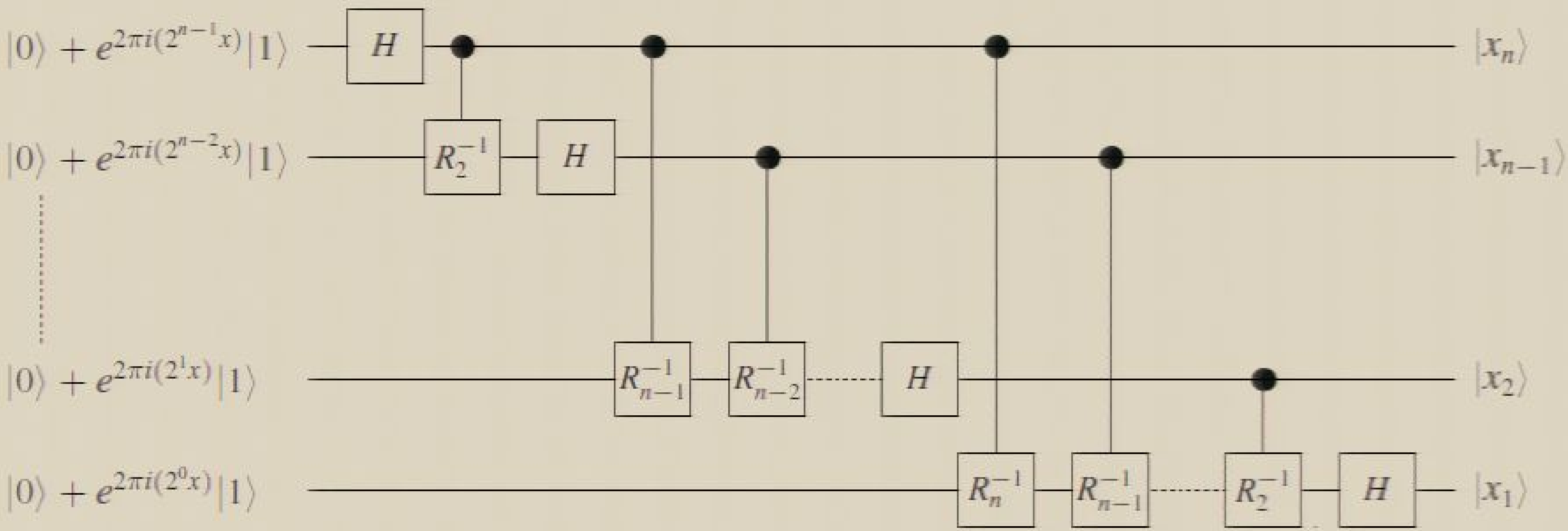
- In the Inverse QFT, qubits are used as quantum control, then measured
- We can replace this with measurement, followed by classical control
- Prepare the qubits of the output register one at a time
- Start with x_n and work down to x_1
- Use these results to classically control the phase rotation corrections

Circuit for sampling individual bit

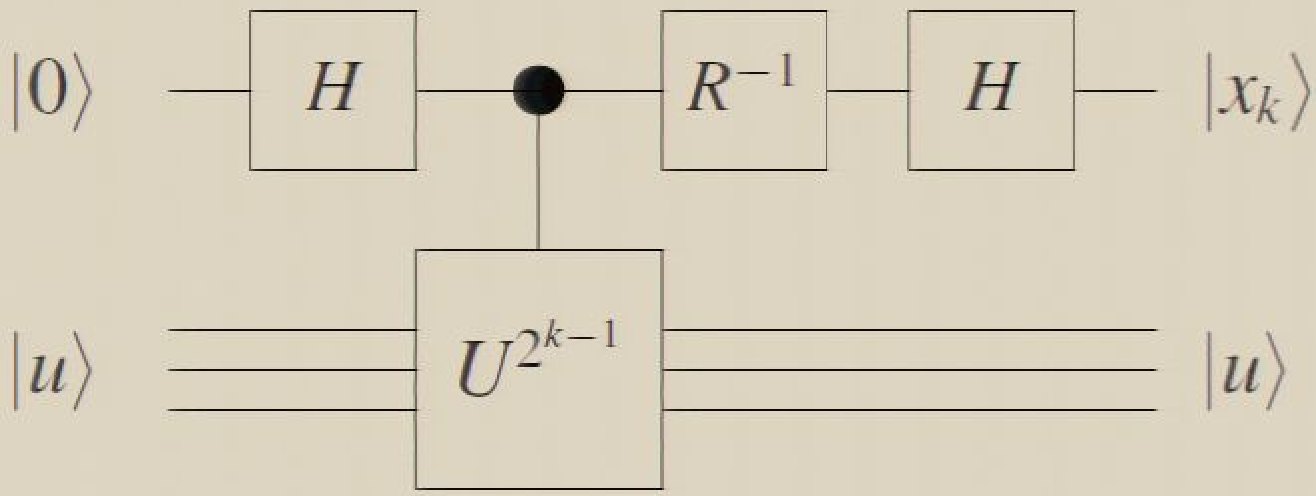


- In general, R estimates and tries to correct the phase shift $e^{2\pi i(2^{k-1}x-(0.x_k))}$
- We can improve the way we estimate R
- Example: multiple measurements for each bit x_k

Phase Estimation with Inverse QFT



Circuit for sampling individual bit



- In general, R estimates and tries to correct the phase shift $e^{2\pi i(2^{k-1}x-(0.x_k))}$
- We can improve the way we estimate R
- Example: multiple measurements for each bit x_k

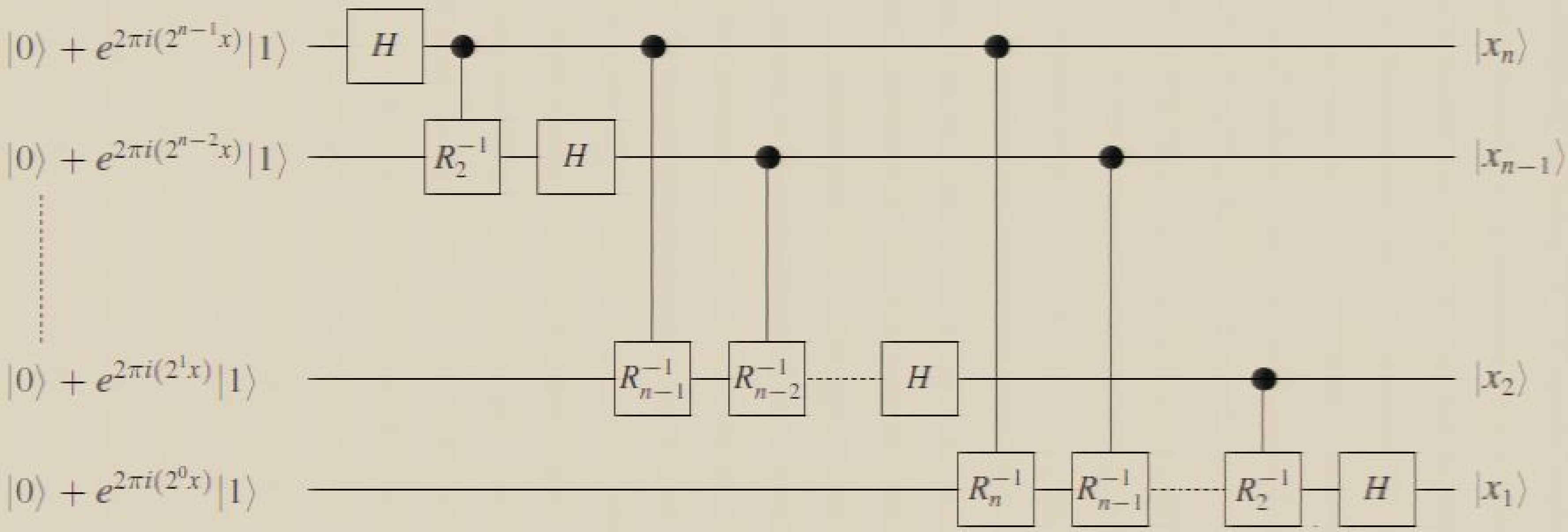
Approximate QFT

- We can also relax the way we estimate R
- AQFT idea introduced by Coppersmith (1994)
- QFT uses controlled phase rotation gates with some very small angles
- Disregarding the smallest ones should not significantly affect the result
- AQFT can be parameterized by "depth" m, giving AQFT_m
- Controlled phase rotations of less than $e^{2\pi i(2^{-m})}$ are removed

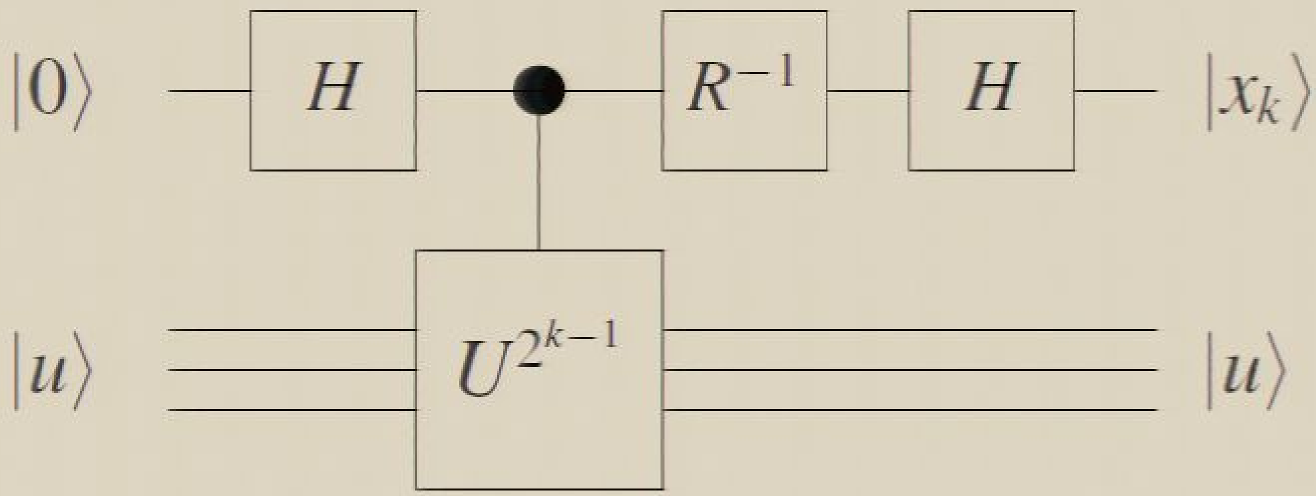
Semi-classical Phase Estimation

- In the Inverse QFT, qubits are used as quantum control, then measured
- We can replace this with measurement, followed by classical control
- Prepare the qubits of the output register one at a time
- Start with x_n and work down to x_1
- Use these results to classically control the phase rotation corrections

Phase Estimation with Inverse QFT



Circuit for sampling individual bit

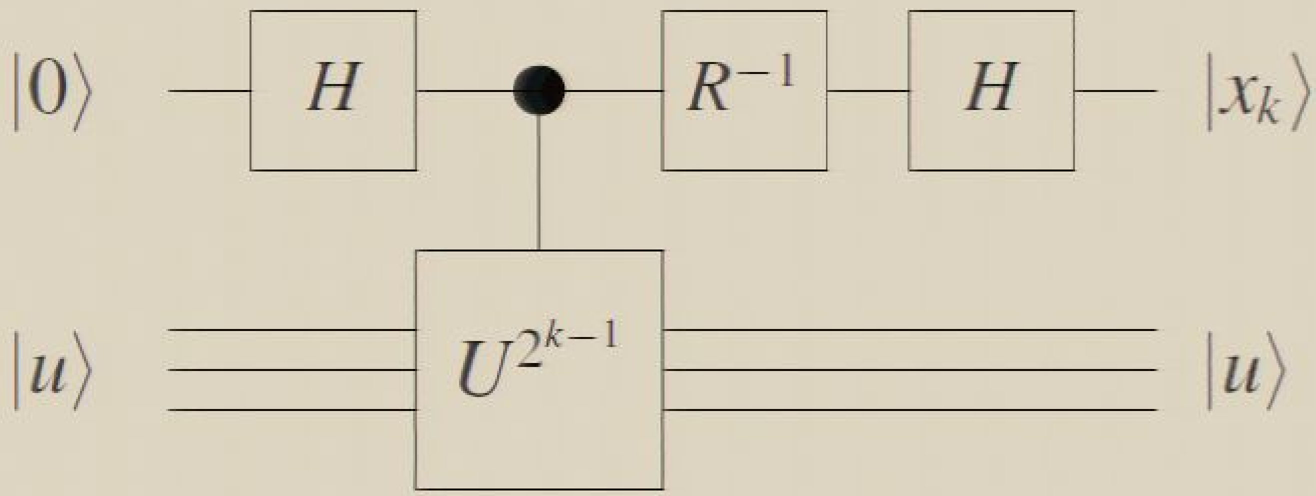


- In general, R estimates and tries to correct the phase shift $e^{2\pi i(2^{k-1}x-(0.x_k))}$
- We can improve the way we estimate R
- Example: multiple measurements for each bit x_k

Approximate QFT

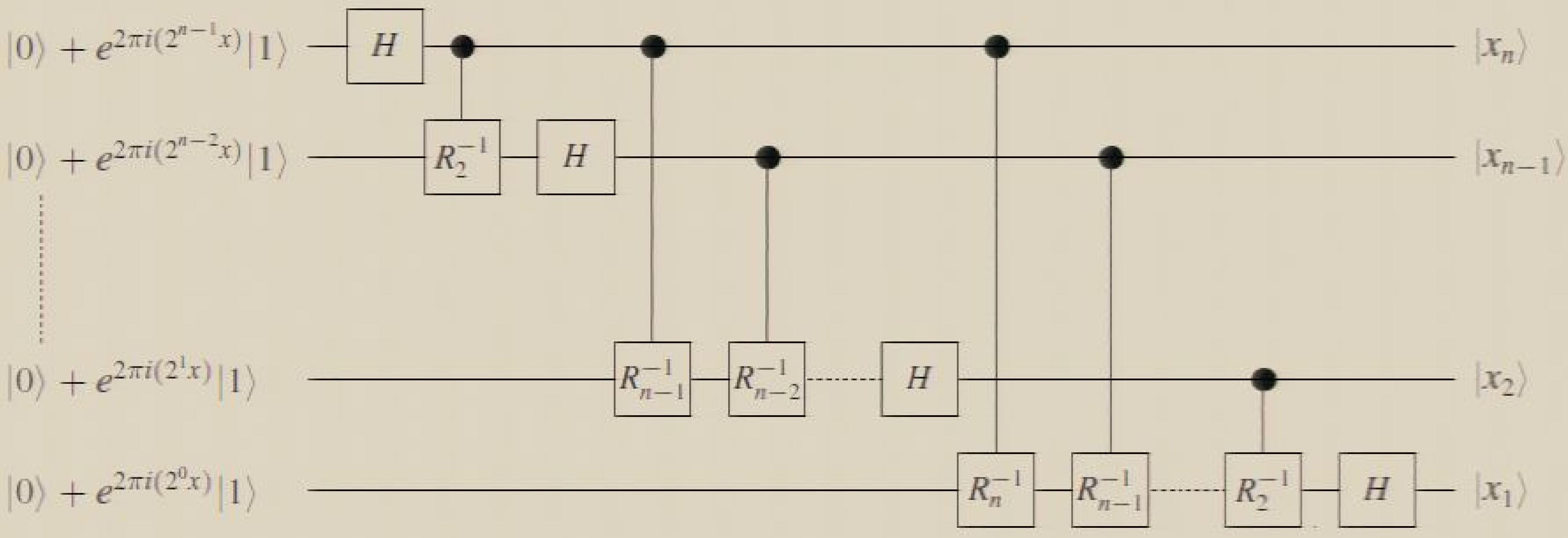
- We can also relax the way we estimate R
- AQFT idea introduced by Coppersmith (1994)
- QFT uses controlled phase rotation gates with some very small angles
- Disregarding the smallest ones should not significantly affect the result
- AQFT can be parameterized by “depth” m, giving AQFT_m
- Controlled phase rotations of less than $e^{2\pi i(2^{-m})}$ are removed

Circuit for sampling individual bit



- In general, R estimates and tries to correct the phase shift $e^{2\pi i(2^{k-1}x-(0.x_k))}$
- We can improve the way we estimate R
- Example: multiple measurements for each bit x_k

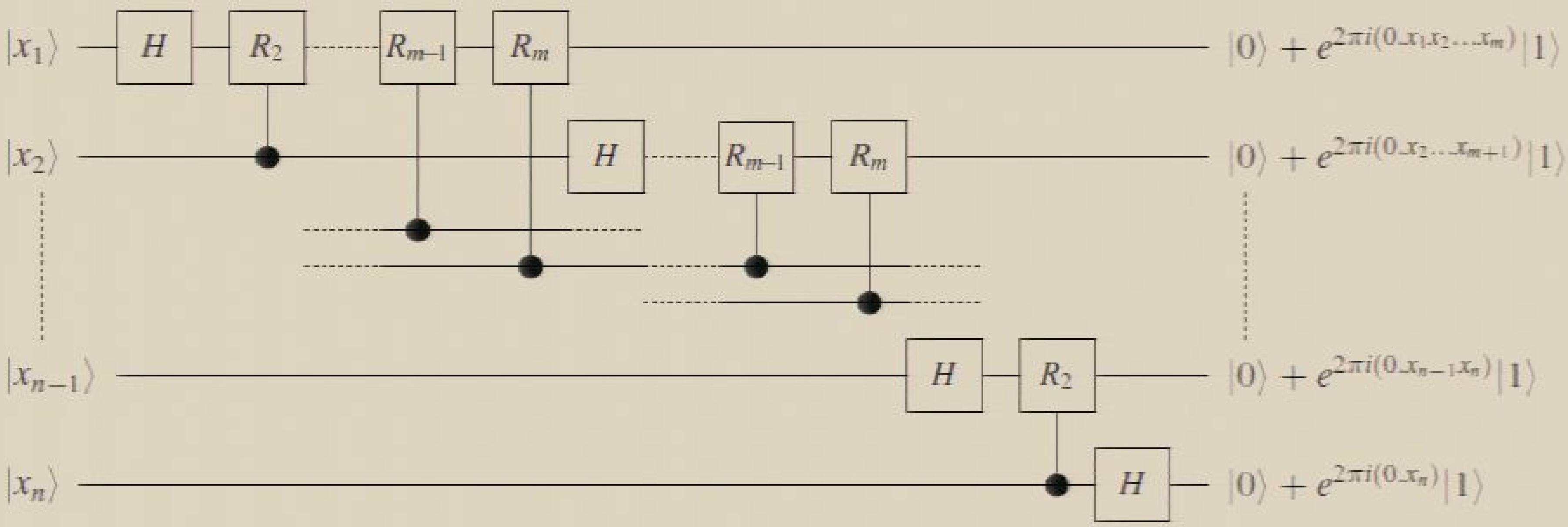
Phase Estimation with Inverse QFT



Approximate QFT

- We can also relax the way we estimate R
- AQFT idea introduced by Coppersmith (1994)
- QFT uses controlled phase rotation gates with some very small angles
- Disregarding the smallest ones should not significantly affect the result
- AQFT can be parameterized by “depth” m, giving AQFT_m
- Controlled phase rotations of less than $e^{2\pi i(2^{-m})}$ are removed

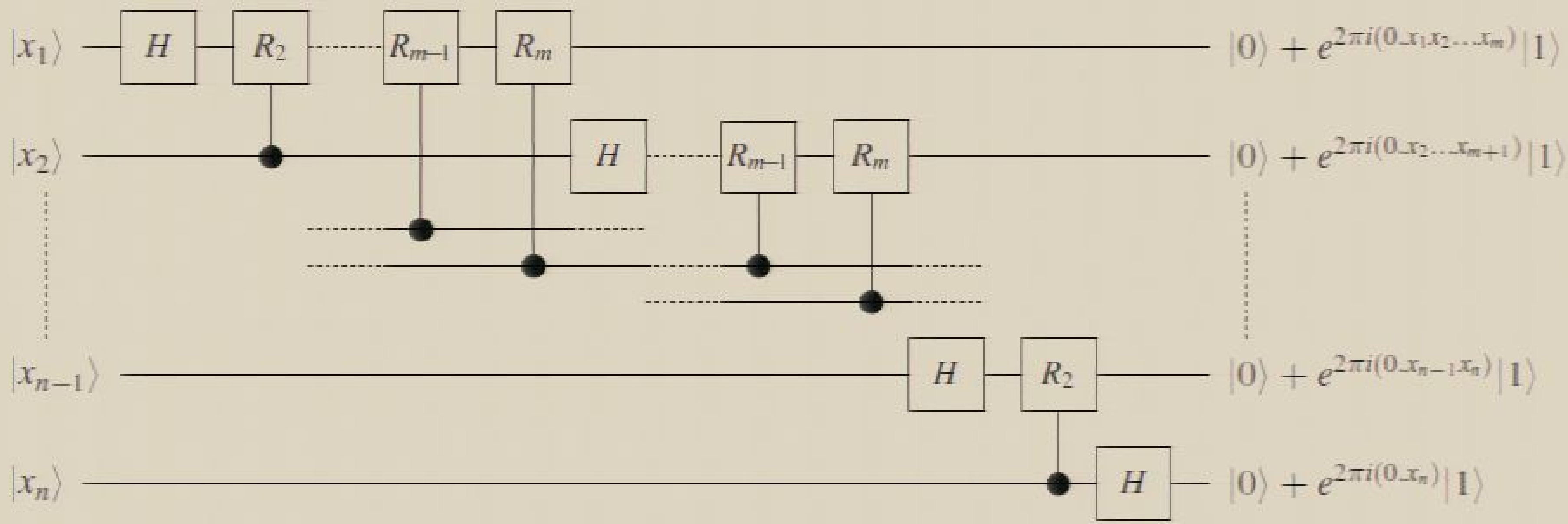
Approximate QFT Circuit



Generalized Approximate QFT

- AQFT_m can be generalized to any system of estimating R of equivalent accuracy
- Let P be the probability that $|x - \hat{x}| \leq 2^{-(n+1)}$
- P is the probability that $\hat{x}$ is the nearest fractional estimate of x

Approximate QFT Circuit



Outline

1. Review of Phase Estimation and AQFT
   - Phase Estimation
   - Approximate Quantum Fourier Transform
2. Classical Post-processing
   - Maximum Likelihood Estimation
   - Classical Post-processing Algorithm
3. Hidden Subgroup Problems
   - Definitions
   - Dihedral HSP
   - Postprocessing for Dihedral HSP

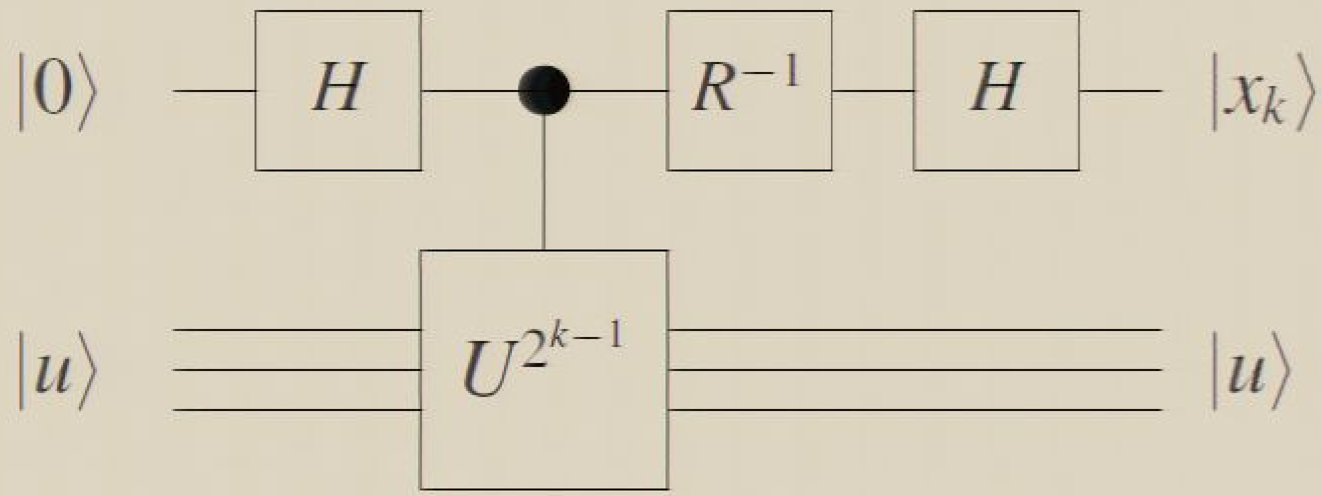
Approximate QFT

- We can also relax the way we estimate R
- AQFT idea introduced by Coppersmith (1994)
- QFT uses controlled phase rotation gates with some very small angles
- Disregarding the smallest ones should not significantly affect the result
- AQFT can be parameterized by "depth" m, giving AQFT_m
- Controlled phase rotations of less than $e^{2\pi i(2^{-m})}$ are removed

Generalized Approximate QFT

- AQFT_m can be generalized to any system of estimating R of equivalent accuracy
- Let P be the probability that $|x - \hat{x}| \leq 2^{-(n+1)}$
- P is the probability that $\hat{x}$ is the nearest fractional estimate of x

Circuit for sampling individual bit



- In general, R estimates and tries to correct the phase shift $e^{2\pi i(2^{k-1}x-(0.x_k))}$
- We can improve the way we estimate R
- Example: multiple measurements for each bit x_k

Approximate QFT

- We can also relax the way we estimate R
- AQFT idea introduced by Coppersmith (1994)
- QFT uses controlled phase rotation gates with some very small angles
- Disregarding the smallest ones should not significantly affect the result
- AQFT can be parameterized by "depth" m, giving AQFT_m
- Controlled phase rotations of less than $e^{2\pi i(2^{-m})}$ are removed

Log-depth Approximate QFT

- Given log-depth AQFT with $m \geq \log_2 n + 2$
- We have a lower bound $P \geq \frac{4}{\pi^2} - \frac{1}{4n}$
- Compare with $P \geq \frac{4}{\pi^2}$ for full QFT
- For large n, difference between QFT and log-depth AQFT is negligible

Generalized Approximate QFT

- AQFT_m can be generalized to any system of estimating R of equivalent accuracy
- Let P be the probability that $|x - \hat{x}| \leq 2^{-(n+1)}$
- P is the probability that $\hat{x}$ is the nearest fractional estimate of x

Log-depth Approximate QFT

- Given log-depth AQFT with $m \geq \log_2 n + 2$
- We have a lower bound $P \geq \frac{4}{\pi^2} - \frac{1}{4n}$
- Compare with $P \geq \frac{4}{\pi^2}$ for full QFT
- For large n, difference between QFT and log-depth AQFT is negligible

Low-depth Approximate AQFT

- We are interested in the case where $m < O(\log_2 n)$
- Measurement results for bits x_k no longer give a good estimate with significant probability
- Kitaev gives a phase estimation algorithm based on low-depth AQFT
- Idea: Sample x_k $O(\log n)$ times to estimate $x_k.x_{k+1}x_{k+2}\ldots$
- Combine information from all the sampled bits to determine phase

Log-depth Approximate QFT

- Given log-depth AQFT with $m \geq \log_2 n + 2$
- We have a lower bound $P \geq \frac{4}{\pi^2} - \frac{1}{4n}$
- Compare with $P \geq \frac{4}{\pi^2}$ for full QFT
- For large n, difference between QFT and log-depth AQFT is negligible

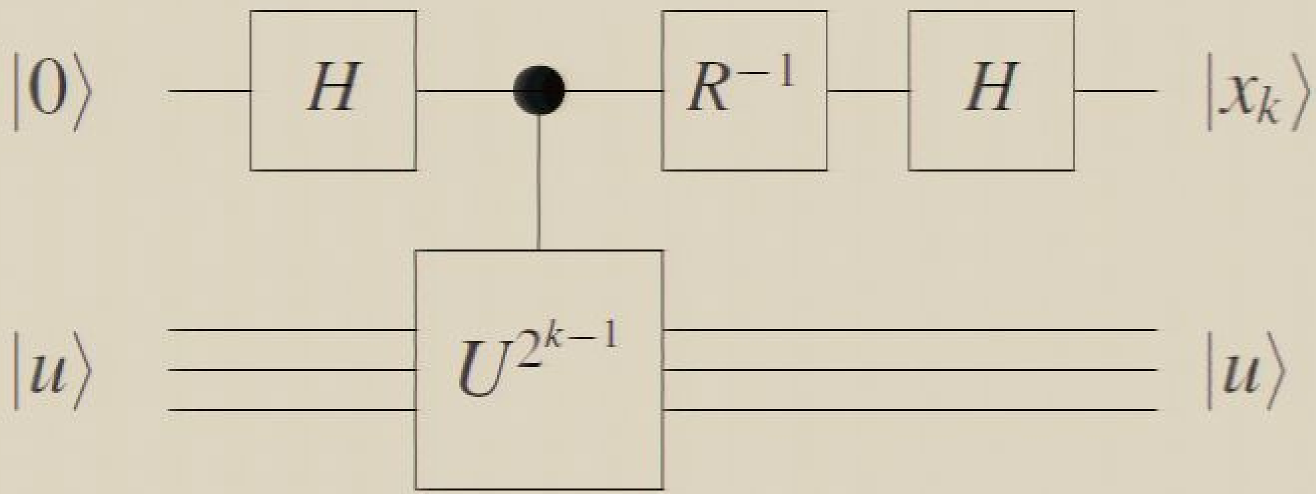
Low-depth Approximate AQFT

- We are interested in the case where $m < O(\log_2 n)$
- Measurement results for bits x_k no longer give a good estimate with significant probability
- Kitaev gives a phase estimation algorithm based on low-depth AQFT
- Idea: Sample x_k $O(\log n)$ times to estimate $x_k.x_{k+1}x_{k+2}\dots$
- Combine information from all the sampled bits to determine phase

Approximate QFT

- We can also relax the way we estimate R
- AQFT idea introduced by Coppersmith (1994)
- QFT uses controlled phase rotation gates with some very small angles
- Disregarding the smallest ones should not significantly affect the result
- AQFT can be parameterized by “depth” m, giving AQFT_m
- Controlled phase rotations of less than $e^{2\pi i(2^{-m})}$ are removed

Circuit for sampling individual bit



- In general, R estimates and tries to correct the phase shift $e^{2\pi i(2^{k-1}x-(0.x_k))}$
- We can improve the way we estimate R
- Example: multiple measurements for each bit x_k

Approximate QFT

- We can also relax the way we estimate R
- AQFT idea introduced by Coppersmith (1994)
- QFT uses controlled phase rotation gates with some very small angles
- Disregarding the smallest ones should not significantly affect the result
- AQFT can be parameterized by "depth" m, giving AQFT_m
- Controlled phase rotations of less than $e^{2\pi i(2^{-m})}$ are removed

Log-depth Approximate QFT

- Given log-depth AQFT with $m \geq \log_2 n + 2$
- We have a lower bound $P \geq \frac{4}{\pi^2} - \frac{1}{4n}$
- Compare with $P \geq \frac{4}{\pi^2}$ for full QFT
- For large n, difference between QFT and log-depth AQFT is negligible

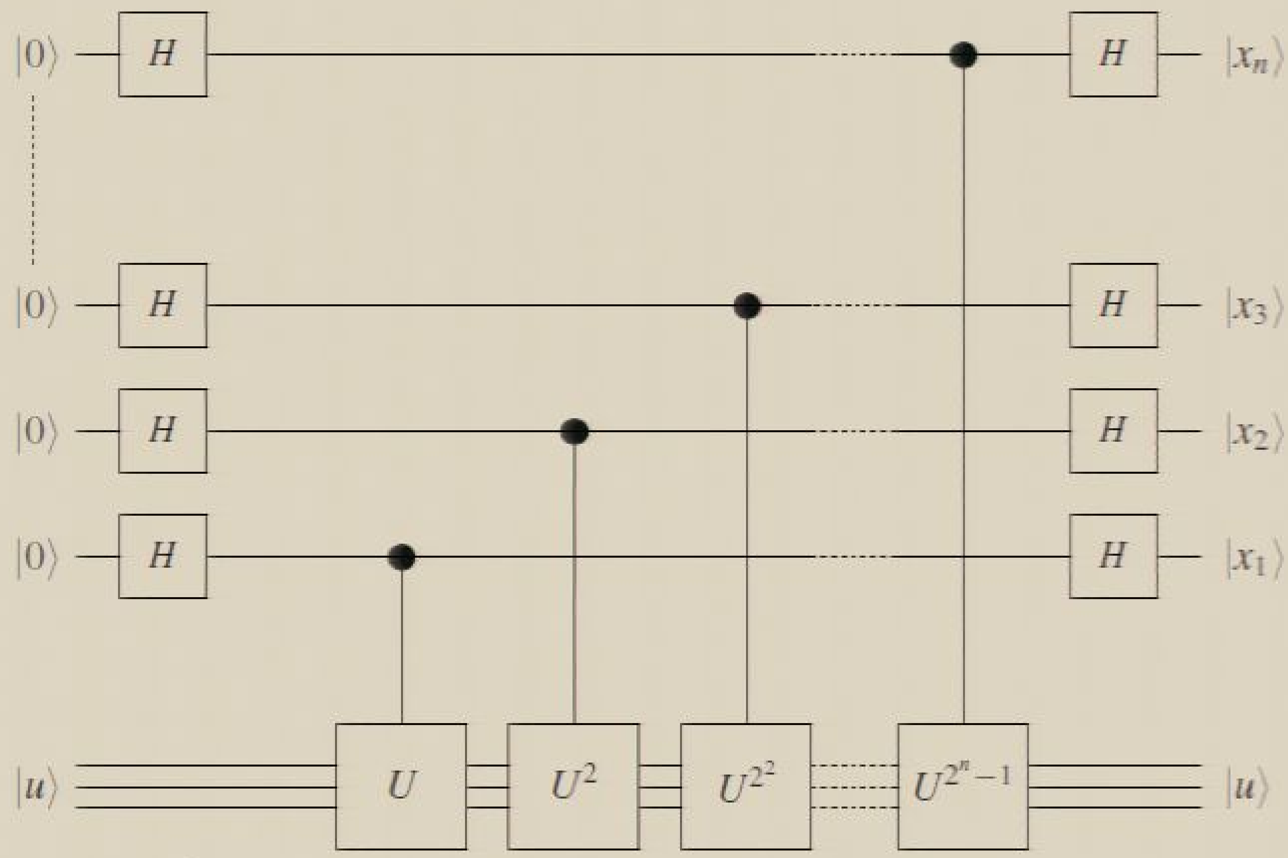
Low-depth Approximate AQFT

- We are interested in the case where $m < O(\log_2 n)$
- Measurement results for bits x_k no longer give a good estimate with significant probability
- Kitaev gives a phase estimation algorithm based on low-depth AQFT
- Idea: Sample x_k $O(\log n)$ times to estimate $x_k.x_{k+1}x_{k+2}\ldots$
- Combine information from all the sampled bits to determine phase

Low-depth Approximate AQFT

- We are interested in the case where $m < O(\log_2 n)$
- Measurement results for bits x_k no longer give a good estimate with significant probability
- Kitaev gives a phase estimation algorithm based on low-depth AQFT
- Idea: Sample x_k $O(\log n)$ times to estimate $x_k.x_{k+1}x_{k+2}\ldots$
- Combine information from all the sampled bits to determine phase

Kitaev's Phase Estimation Algorithm



Outline

- Phase Estimation
- Approximate Quantum Fourier Transform

2. Classical Post-processing
   - Maximum Likelihood Estimation
   - Classical Post-processing Algorithm

- Definitions
- Dihedral HSP
- Postprocessing for Dihedral HSP

Maximum Likelihood Estimation

- There is a better way of "combining" information from all measurements
- Method of Maximum Likelihood gives estimate with optimal statistical properties
- Given a fixed set of outcomes
- The likelihood of a particular input is the probability of obtaining the fixed outcome given that input
- Compare with probability: we are given a fixed input, and consider the distribution of outcomes

Maximum Likelihood Estimation

- Given fixed outcomes, construct a Likelihood function $L(x)$ over the set of possible inputs
- The Maximum Likelihood Estimate is the input $\hat{x}$ which maximizes L
- The correct input x should be among those with highest likelihood
- This depends on the internal consistency of the data
- Also depends on how well the data distinguishes the correct input x from other inputs

Likelihood Function for Phase Estimation

- Let r_k be the phase correction applied by R.
  $R : |0\rangle \mapsto |0\rangle$, and $R : |1\rangle \mapsto e^{2\pi i r_k}|1\rangle$
- We measure bit x_k with result $|0\rangle$ c_k times, and $|1\rangle$ s_k times
- Probability of obtaining this result from input x (ignoring constant) is

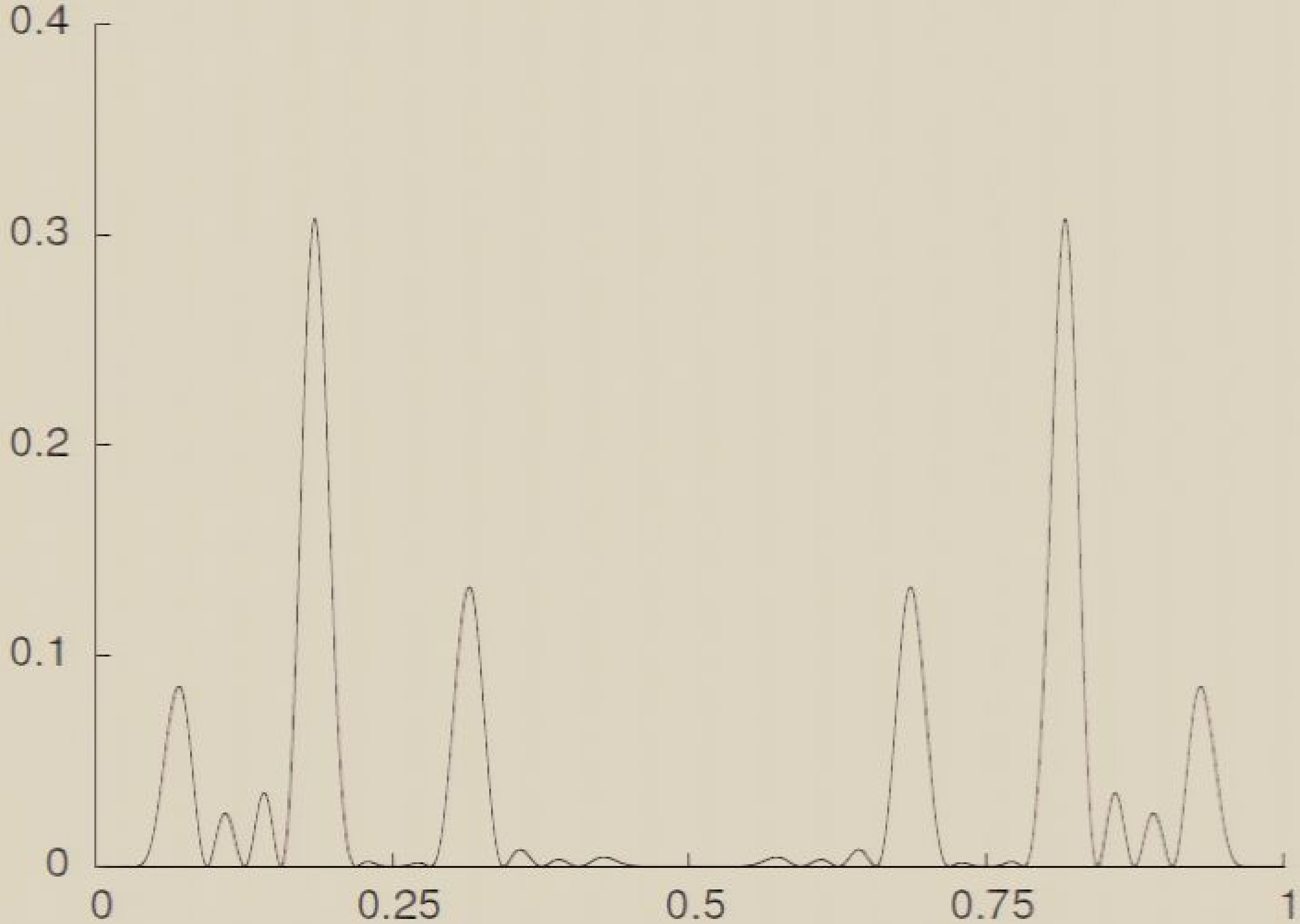
$$f_k(x) = \left(\cos^2 \pi(2^{k-1}x - r_k)\right)^{c_k} \left(\sin^2 \pi(2^{k-1}x - r_k)\right)^{s_k}$$

- Likelihood function for outcomes from all measurements is

$$L(x) = \prod_{k=1}^{n} f_k(x)$$

- With at least 2^{n-1} distinct roots, direct analysis is infeasible

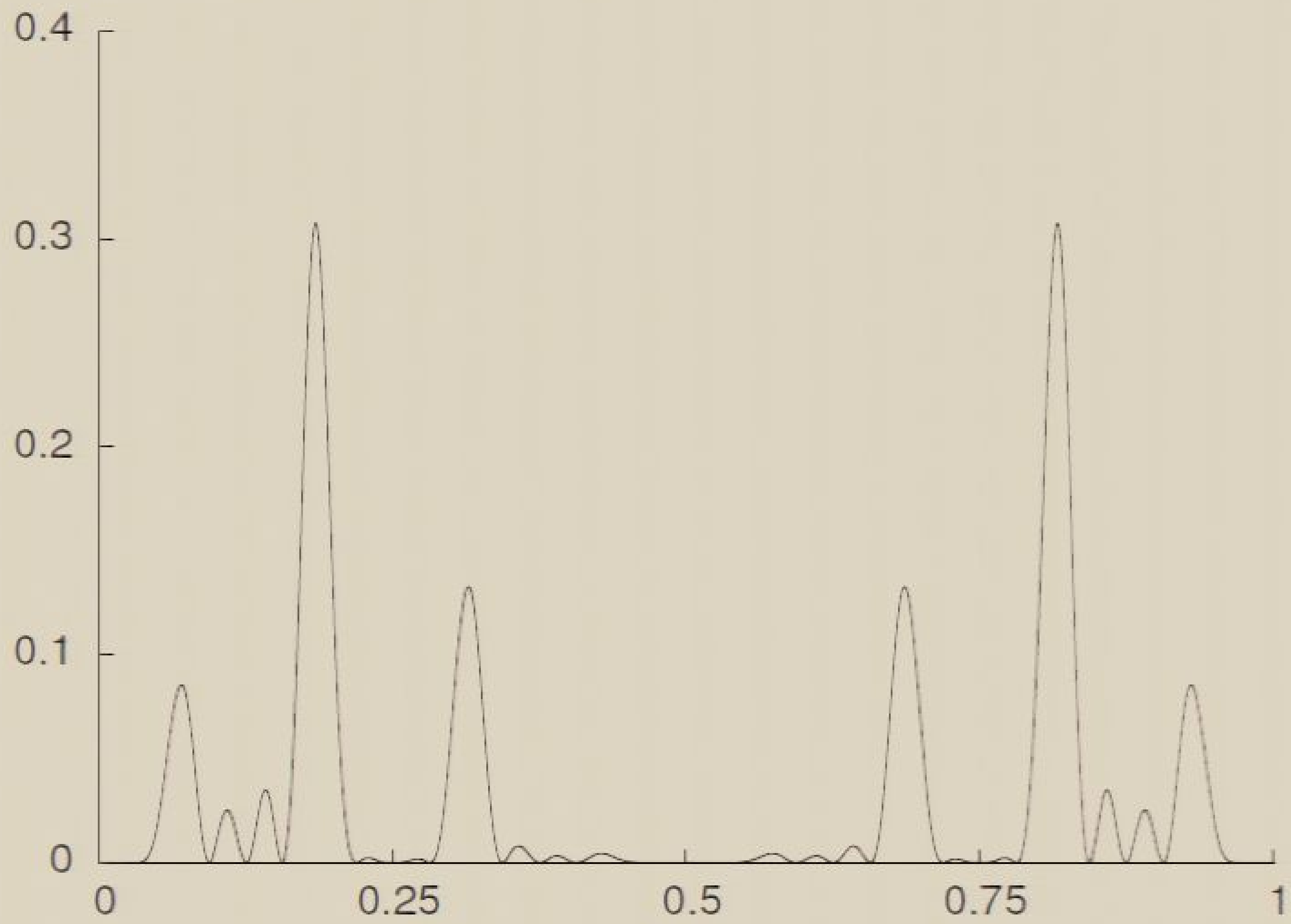
Example Likelihood Function



Post-processing Strategy

- Try to find intervals where $L(x)$ can be easily bounded
- Find regions where we are unlikely to find the largest maxima globally
- Use step functions to bound $L(x)$
- Focus attention on refining the bound for tall steps in the step function

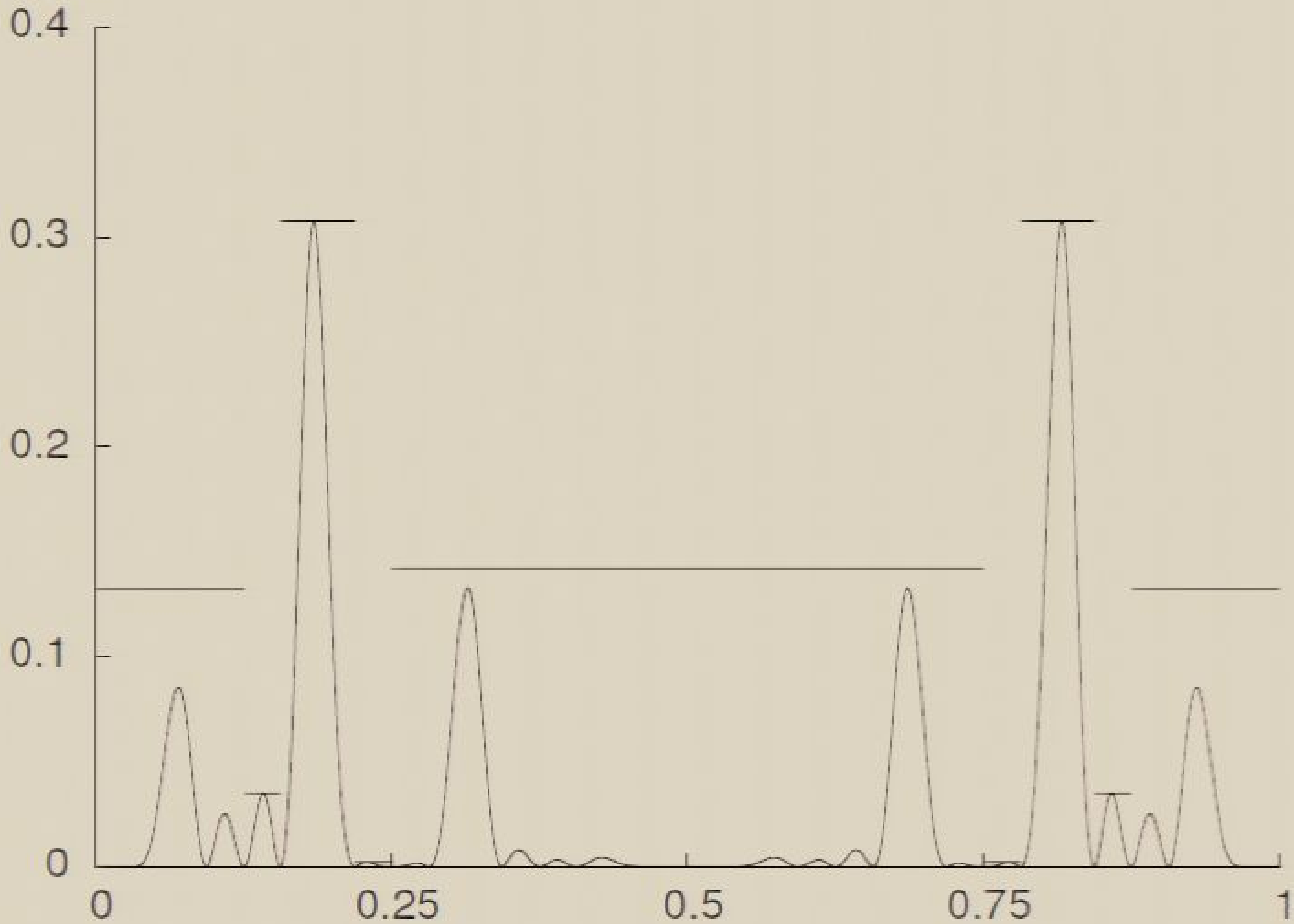
Example Likelihood Function



Post-processing Strategy

- Try to find intervals where $L(x)$ can be easily bounded
- Find regions where we are unlikely to find the largest maxima globally
- Use step functions to bound $L(x)$
- Focus attention on refining the bound for tall steps in the step function

Likelihood Function Bounded by Step Function



How to Find Bounds for $L(x)$

- Split $L(x)$ into subfunctions:

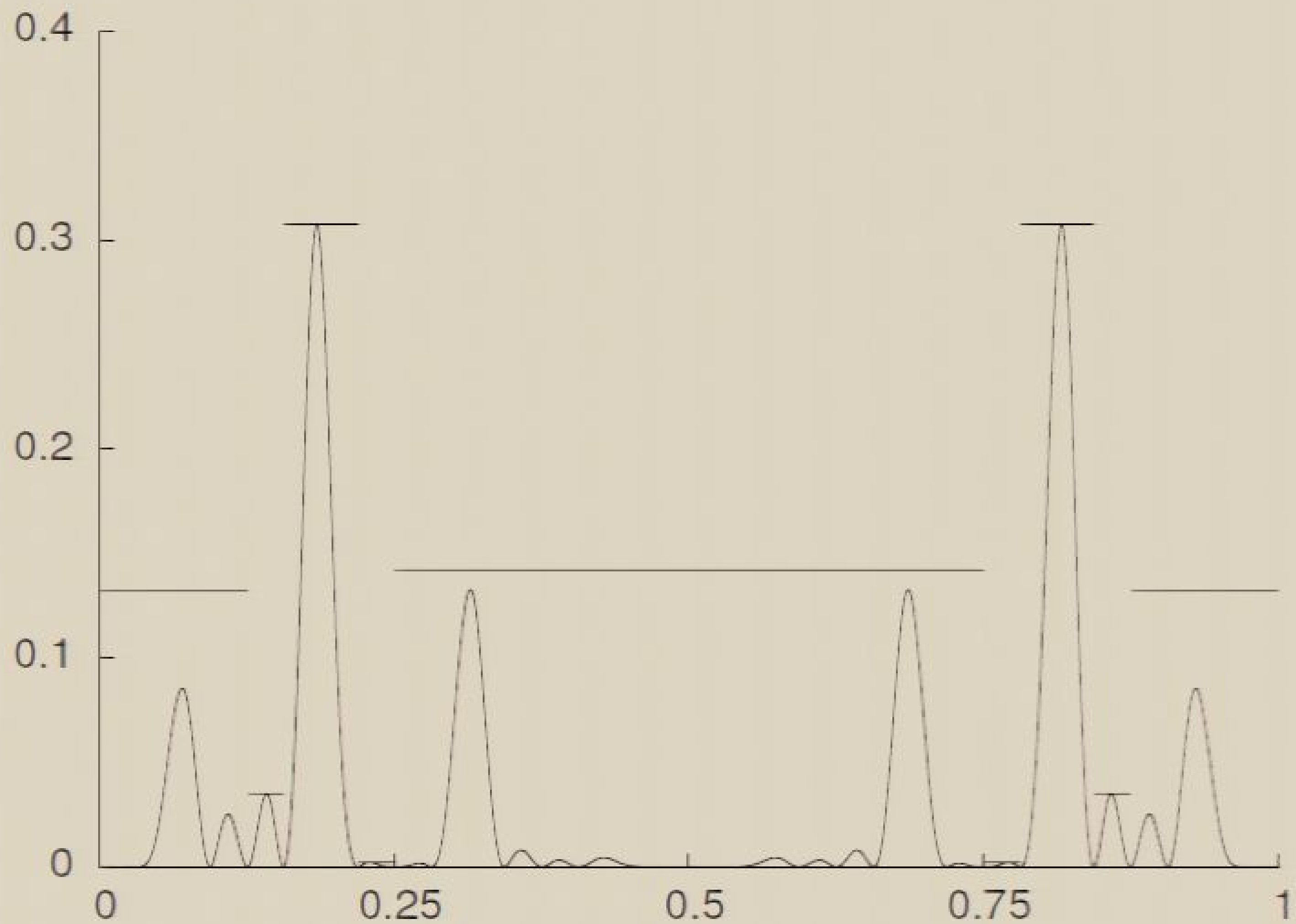
$$L_j(x) = \prod_{k=1}^{j} f_k(x)$$

and

$$S_j(x) = \prod_{k=j+1}^{n} f_k(x)$$

- A step bound consists of a bound on $L_j(x)$ combined with a bound on $S_j(x)$

Likelihood Function Bounded by Step Function



How to Find Bounds for $L(x)$

- Split $L(x)$ into subfunctions:

$$L_j(x) = \prod_{k=1}^{j} f_k(x)$$

  and

$$S_j(x) = \prod_{k=j+1}^{n} f_k(x)$$

- A step bound consists of a bound on $L_j(x)$ combined with a bound on $S_j(x)$

Likelihood Function for Phase Estimation

- Let r_k be the phase correction applied by R.
  $R : |0\rangle \mapsto |0\rangle$, and $R : |1\rangle \mapsto e^{2\pi i r_k}|1\rangle$
- We measure bit x_k with result $|0\rangle$ c_k times, and $|1\rangle$ s_k times
- Probability of obtaining this result from input x (ignoring constant) is

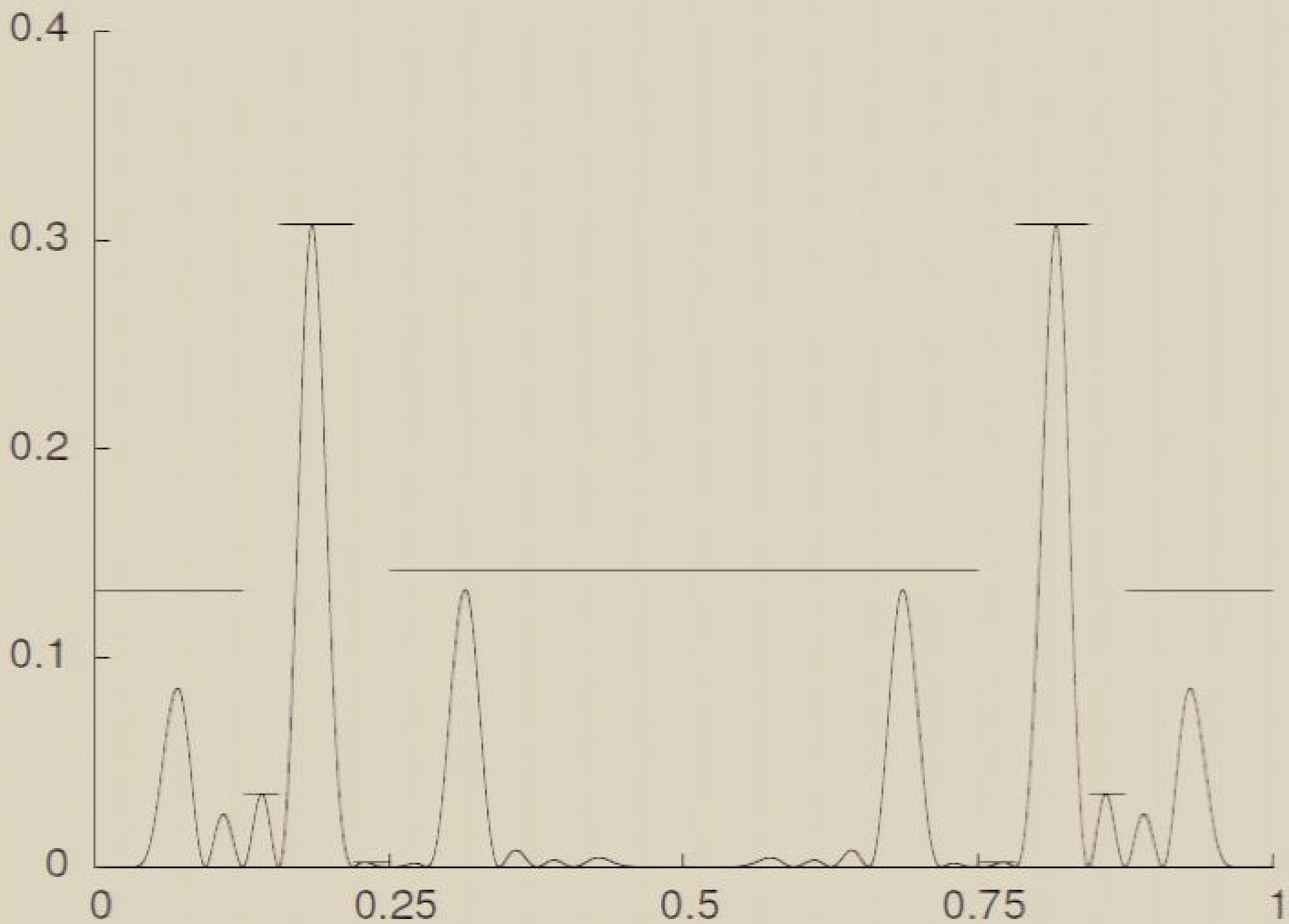
$$f_k(x) = \left(\cos^2 \pi(2^{k-1}x - r_k)\right)^{c_k} \left(\sin^2 \pi(2^{k-1}x - r_k)\right)^{s_k}$$

- Likelihood function for outcomes from all measurements is

$$L(x) = \prod_{k=1}^{n} f_k(x)$$

- With at least 2^{n-1}distinct roots, direct analysis is infeasible

Likelihood Function Bounded by Step Function



How to Find Bounds for $L(x)$

- Split $L(x)$ into subfunctions:

$$L_j(x) = \prod_{k=1}^{j} f_k(x)$$

and

$$S_j(x) = \prod_{k=j+1}^{n} f_k(x)$$

- A step bound consists of a bound on $L_j(x)$ combined with a bound on $S_j(x)$

How to Find Bounds for $L(x)$

- Given a bound for $L(x)$ in the form of a step function
- Find the tallest step and try to replace it with better bounds
- If it's tight, we've found a maximum
- We start with $L(x) \leq 1$ on the entire interval

How to Find Bounds for $L(x)$

- Split $L(x)$ into subfunctions:

$$L_j(x) = \prod_{k=1}^{j} f_k(x)$$

and

$$S_j(x) = \prod_{k=j+1}^{n} f_k(x)$$

- A step bound consists of a bound on $L_j(x)$ combined with a bound on $S_j(x)$

How to Find Bounds for $L(x)$

- Given a bound for $L(x)$ in the form of a step function
- Find the tallest step and try to replace it with better bounds
- If it's tight, we've found a maximum
- We start with $L(x) \leq 1$ on the entire interval

How to Find Bounds for $L(x)$

- Split $L(x)$ into subfunctions:

$$L_j(x) = \prod_{k=1}^{j} f_k(x)$$

and

$$S_j(x) = \prod_{k=j+1}^{n} f_k(x)$$

- A step bound consists of a bound on $L_j(x)$ combined with a bound on $S_j(x)$

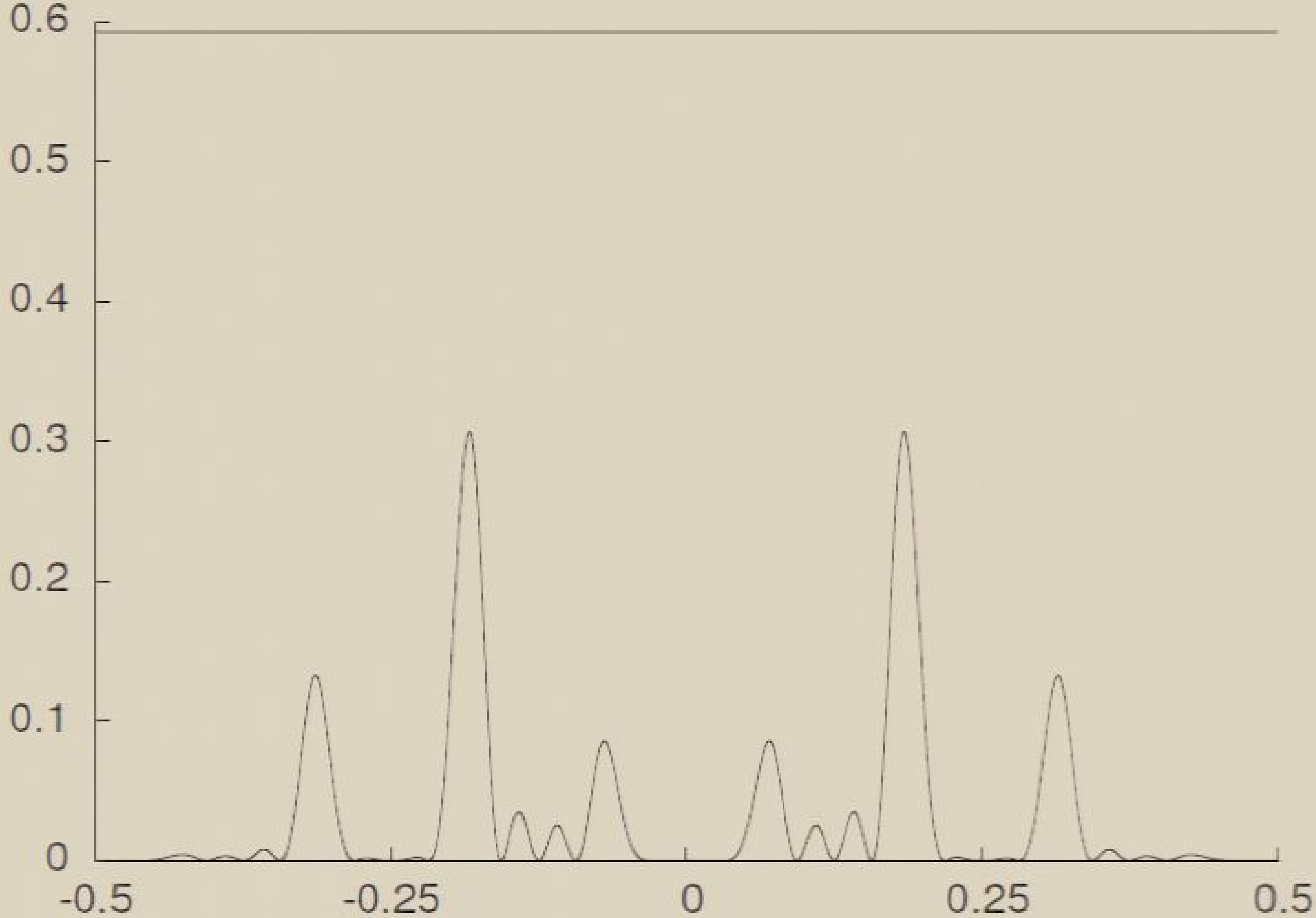
Interval Update Procedure

- Store step functions as a collection of intervals and bounds
- Also store information about which subfunctions $L_j(x)S_j(x)$ the bound was derived from
- Each step bound derived from bounds on $L_j(x)S_j(x)$ will have consecutive roots of $L_j(x)$ as endpoints
- Simplest update procedure: Take tallest step, replace it with steps derived from $L_{j+1}(x)S_{j+1}(x)$
- If $j = n$ we have a bound between consecutive roots of $L(x)$

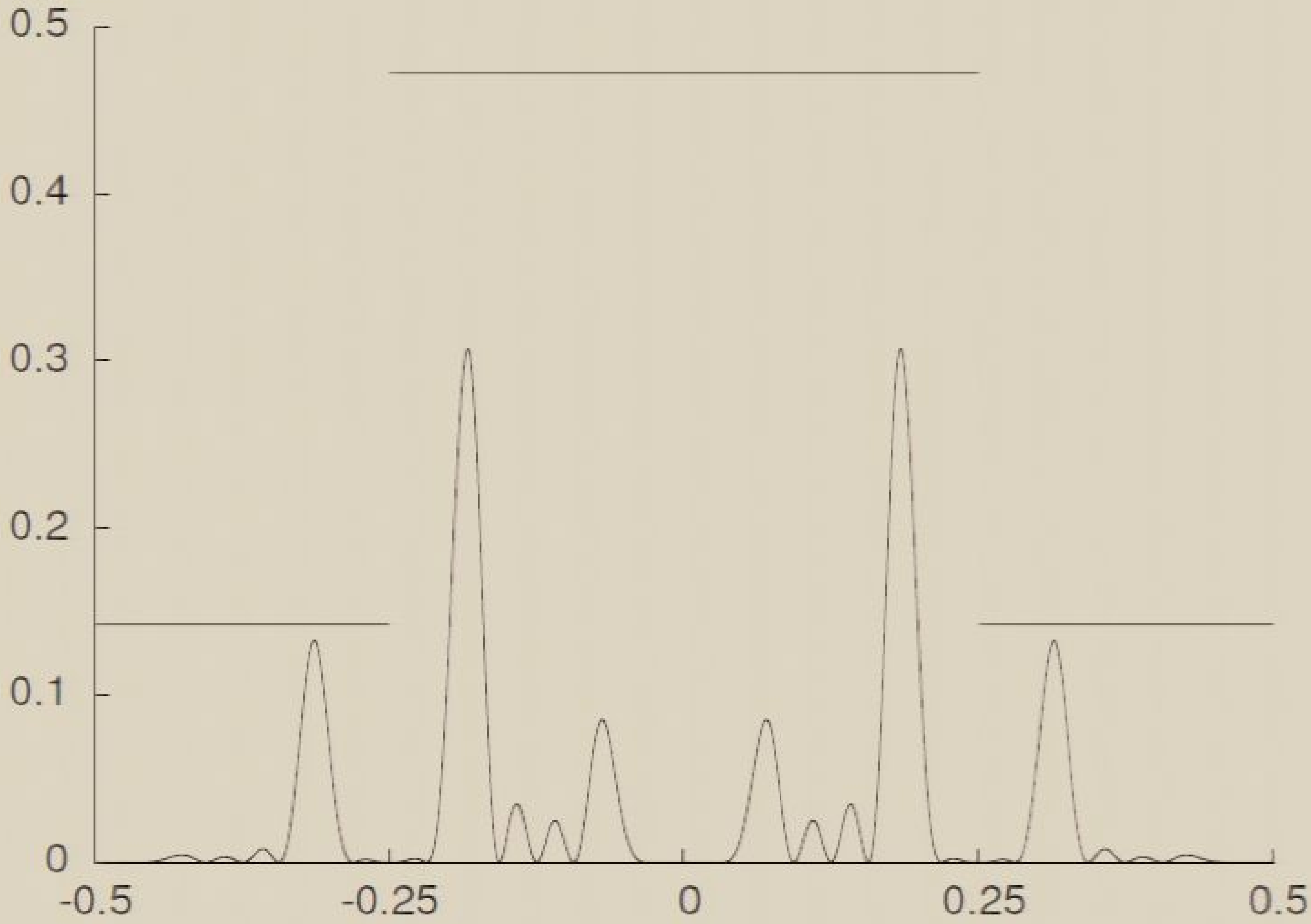
Sample Algorithm Execution

- $L(x) = \cos^2(\pi x) \sin^2(2\pi x) \sin^2(2^2\pi x) \sin^2(2^3\pi x) \cos^2(2^4\pi x)$
- Start with bound $L(x) \leq 1$ on interval $\mathbb{R}$
- Use trivial bounds $S_j(x) \leq 1$

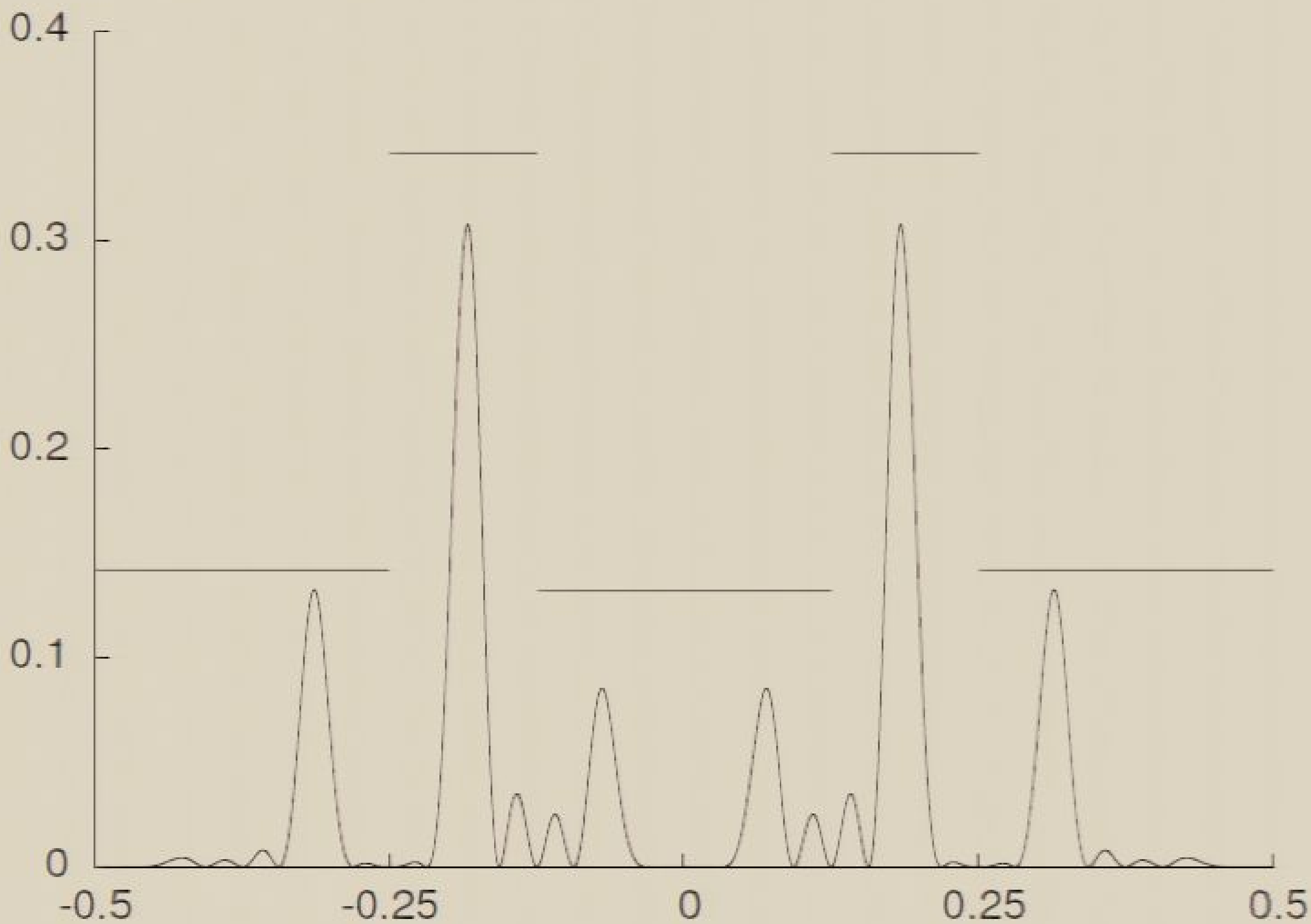
A Step Function Bound for $L(x)$



A Step Function Bound for $L(x)$



A Step Function Bound for $L(x)$



Improvement on Interval Update

- We can use the interval update procedure to build bounds for $S_j(2^{-j}x)$
- Fix the number of steps in an estimate as a parameter
- We can improve the bounds in the estimate by updating more intervals
- This procedure can simulate doing several interval updates at once
- If we use too many steps, this can become inaccurate, wasting time and memory

Practical Details

- Instead of using $L(x)$ we use the log likelihood function

$$\begin{aligned}\ell(x) &= \log L(x) \\ &= 2\sum_{k=1}^{n} c_k \log\cos\pi(2^{k-1}x - r_k) + s_k \log\sin\pi(2^{k-1} - r_k).\end{aligned}$$

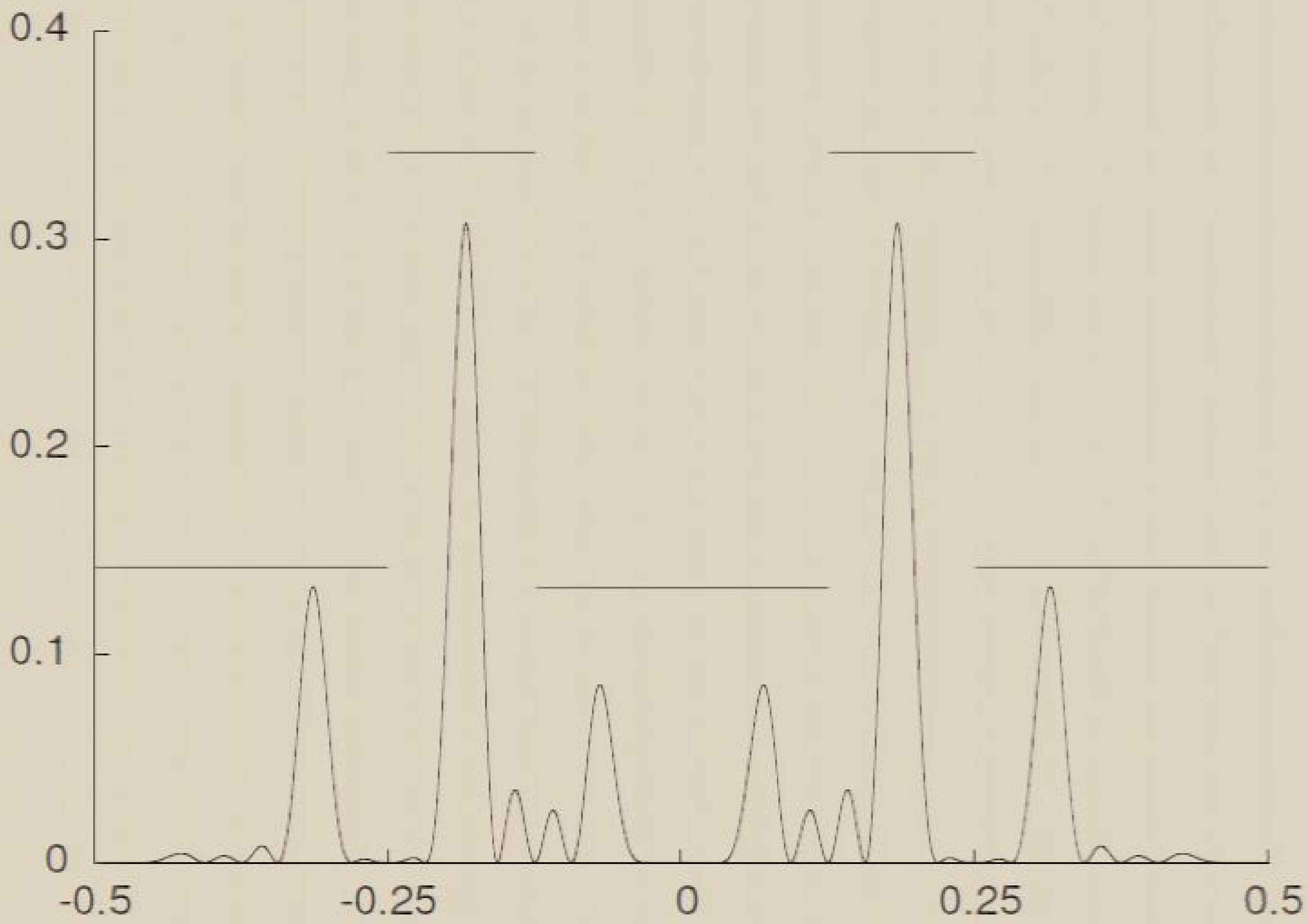
- Addition is faster than multiplication
- Convenient derivative, for bounding subfunction on an interval

$$\ell'(x) = 2\pi\sum_{k=1}^{n} 2^{k-1}\left(-c_k \tan\pi(2^{k-1}x - r_k) + s_k \cot\pi(2^{k-1}x - r_k)\right)$$

Interactive Post-processing for Phase Estimation

- Semi-classical Phase Estimation is interactive
- Measurement results are used to determine phase rotation correction R
- Subfunction $S_j(x)$ is also likelihood function for measurements on bits x_{j+1} to x_n
- Finding rotation corrections can be done while building bounds for $S_j(2^{-j}x)$
- Start with $j = n$, and work down to $j = 1$
- We can also potentially “redo” measurements if a rotation correction does not fit well

A Step Function Bound for $L(x)$



Interactive Post-processing for Phase Estimation

- Semi-classical Phase Estimation is interactive
- Measurement results are used to determine phase rotation correction R
- Subfunction $S_j(x)$ is also likelihood function for measurements on bits x_{j+1} to x_n
- Finding rotation corrections can be done while building bounds for $S_j(2^{-j}x)$
- Start with $j = n$, and work down to $j = 1$
- We can also potentially "redo" measurements if a rotation correction does not fit well

Interactive Post-processing for Phase Estimation

- Semi-classical Phase Estimation is interactive
- Measurement results are used to determine phase rotation correction R
- Subfunction $S_j(x)$ is also likelihood function for measurements on bits x_{j+1} to x_n
- Finding rotation corrections can be done while building bounds for $S_j(2^{-j}x)$
- Start with $j = n$, and work down to $j = 1$
- We can also potentially “redo” measurements if a rotation correction does not fit well

Empirical Results

- Initial implementation, using only basic interval updating
- Run-time depends on quality and consistency of data
- Best case: There exist estimates with likelihood 1
- Algorithm finds these in linear time

Empirical Results

- Initial implementation, using only basic interval updating
- Run-time depends on quality and consistency of data
- Best case: There exist estimates with likelihood 1
- Algorithm finds these in linear time

Outline

1. Review of Phase Estimation and AQFT
   - Phase Estimation
   - Approximate Quantum Fourier Transform

2. Classical Post-processing
   - Maximum Likelihood Estimation
   - Classical Post-processing Algorithm

3. Hidden Subgroup Problems
   - Definitions
   - Dihedral HSP
   - Postprocessing for Dihedral HSP

Empirical Results

- Initial implementation, using only basic interval updating
- Run-time depends on quality and consistency of data
- Best case: There exist estimates with likelihood 1
- Algorithm finds these in linear time