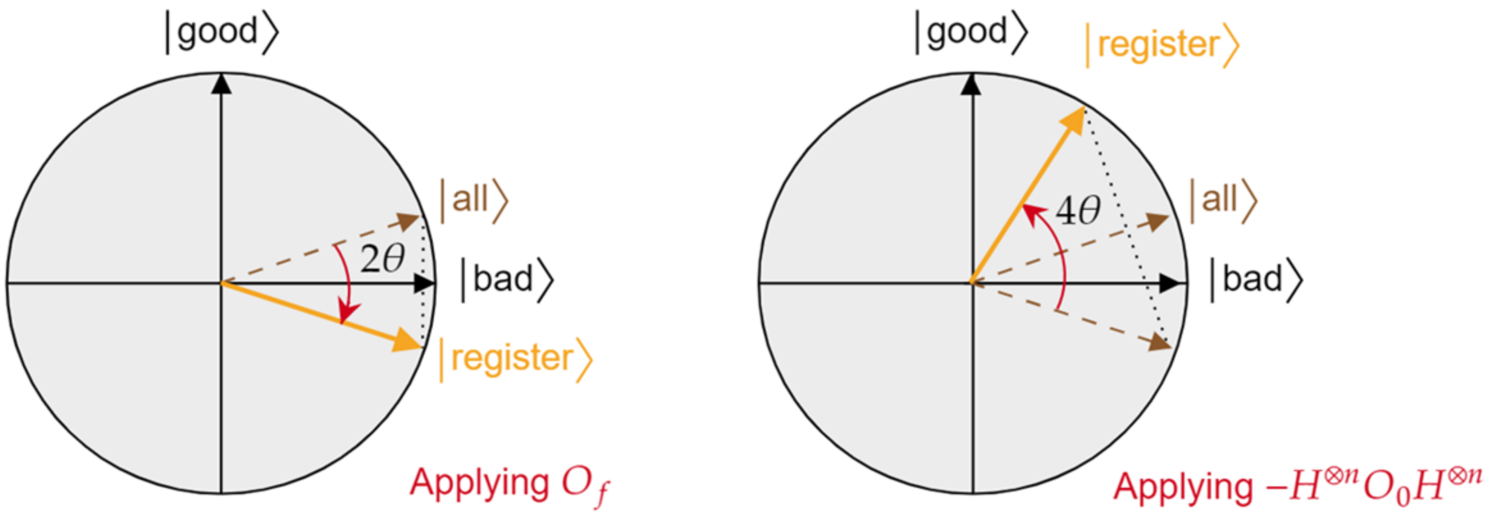
Grover’s search algorithm



Mariia Mykhailova
Principal Software Engineer
Microsoft Quantum Systems

Lecture outline

- Search problem
- Unitaries for Grover’s search
- Grover's search
- Practical aspects of using Grover’s search algorithm

Search problem

The Search Problem

Problem

Given an oracle for $f: \{1, \ldots, N\} \to \{0,1\}$, find an x_0 such that $f(x_0) = 1$ or determine that there is no such x.

Why this problem?

Can encode any problem that allows to check if a given value x is a valid solution:
$f(x) = 1$ if and only if x is a valid solution

- Is x a solution to the travelling salesman problem?
- Is x a solution to the Satisfiability problem?

If we can solve Search, we can find $\min_x f(x)$, or find x such that more conditions hold, e.g., $g(x) = 7$ and $h(x) = 3$.

Example: Satisfiability (SAT)

Given a SAT formula $f: \{0,1\}^n \rightarrow \{0,1\}$, determine if there is an x_0 such that $f(x_0) = 1$
("a satisfying variables assignment").

$$f(x) = \bigwedge_i \bigvee_k y_{ik}, \text{where } y_{ik} = \text{either } x_j \text{ or } \neg x_j$$

For example, $f(x_1, x_2) = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$

This example is 2-SAT (every clause has 2 variables).
2-SAT is easy. k-SAT is NP-complete.

- If we can solve SAT, we can solve other problems we care about (that can be represented as SAT)
- Best classical (or quantum) algorithms run in exponential time

The Search Problem

Problem

Given an oracle for $f: \{1, \dots, N\} \to \{0,1\}$, find an x such that $f(x) = 1$ or determine that there is no such x.

Best classical solution

FOR $x = 1$ to N:

  Evaluate $f(x)$. If $f(x) = 1$, halt and output x.

If the loop ends without finding an x, output "no solution"

Worst case, makes N queries to f.

This is a provably optimal classical solution.

Can we do better with quantum?

The UniqueSearch Problem

For simplicity, let's consider the UniqueSearch problem

You are given an oracle for $f:\{1,\ldots,N\}\rightarrow\{0,1\}$,
and the guarantee that there exists *exactly one* x_0 such that $f(x_0)=1$; find x_0.

- We call x_0 "the marked input"

Quick review: Phase oracles

Phase oracles encode $f(x)$ into the phase of the state

$$U_f\,|x\rangle = (-1)^{f(x)}|x\rangle$$

If $f(x) = 0$, the phase doesn't change

If $f(x) = 1$, the phase is multiplied by -1

$$|x\rangle \quad \boxed{P_f} \quad (-1)^{f(x)}|x\rangle$$

Behavior on superposition states follows from linearity of the oracle:

$$U_f \sum c_i|x_i\rangle = \sum c_i U_f|x_i\rangle = \sum (-1)^{f(x_i)} c_i|x_i\rangle$$

Unitaries for Grover’s search

Unitary 1: Phase oracle

$$|x\rangle \quad \boxed{P_f} \quad (-1)^{f(x)}|x\rangle$$

$$|x\rangle \to \begin{cases} +|x\rangle & \text{if } f(x) = 0 \\ -|x\rangle & \text{if } f(x) = 1 \end{cases}$$

Let's say $N = 2$ and the solution is $x_0 = 1$, i.e., $f(1) = 1$.

$$P_f = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = -2\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = -2|1\rangle\langle 1| + I$$

In general, $P_f = -2|x_0\rangle\langle x_0| + I$

Most Grover's search descriptions use $-P_f = 2|x_0\rangle\langle x_0| - I$

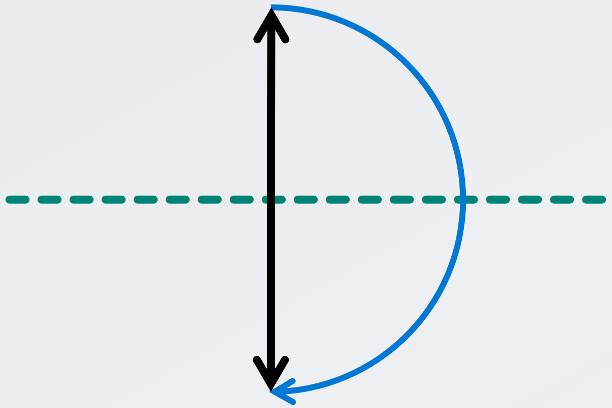
- The global phase -1 doesn't matter in this context
- Allows to represent the oracle as a reflection

Unitary 1 as a reflection

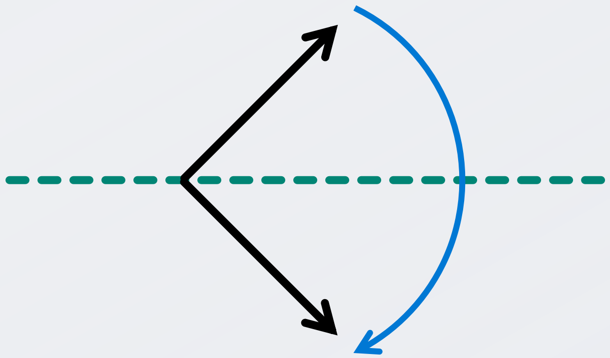
$P_f = 2|\psi\rangle\langle\psi| - I$ is called a **reflection about the state $|\psi\rangle$**:

- leaves $|\psi\rangle$ unchanged: $P_f|\psi\rangle = (2|\psi\rangle\langle\psi| - I)|\psi\rangle = |\psi\rangle$.
- flips phase of states $|\phi\rangle$ orthogonal to $|\psi\rangle$: $P_f|\phi\rangle = (2|\psi\rangle\langle\psi| - I)|\phi\rangle = -|\phi\rangle$

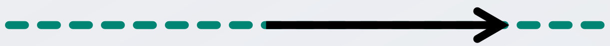
For example, for $|\psi\rangle = |0\rangle$ (represented as horizontal axis) $P_f = Z$



$|1\rangle \rightarrow -|1\rangle$



$0.8|0\rangle + 0.6|1\rangle \rightarrow$
$0.8|0\rangle - 0.6|1\rangle$



$|0\rangle \rightarrow |0\rangle$

Implementing reflections

How to implement $P_f = 2|\psi\rangle\langle\psi| - I$ in general?

If $|\psi\rangle = |1 \dots 1\rangle$, P_f is a multi-controlled Z gate (with -1 global phase):

$$C \cdots CZ = -2|1 \dots 1\rangle\langle 1 \dots 1| + I$$

If we can prepare $|\psi\rangle$, we can implement $P_f = 2|\psi\rangle\langle\psi| - I$

Let $|\psi\rangle = U|11 \dots 1\rangle$ (U is the unitary that prepares $|\psi\rangle$ from the state $|1 \dots 1\rangle$).

Then

$$\begin{aligned} U(C \cdots CZ)U^{-1} &= U(-2|11 \dots 1\rangle\langle 11 \dots 1| + I)U^{-1} = \\ &= -2U|11 \dots 1\rangle\langle 11 \dots 1|U^{-1} + I = \\ &= -2|\psi\rangle\langle\psi| + I = -P_f \end{aligned}$$

We don't use this approach for implementing oracles, since it requires knowing the marked state!
Instead, we use phase oracles implemented via marking oracles

Unitary 2: Reflection about the mean

“The mean” is the equal superposition of all basis states:

$$|\psi_{\text{all}}\rangle = \frac{1}{\sqrt{N}} \sum_{x=1}^{N} |x\rangle$$

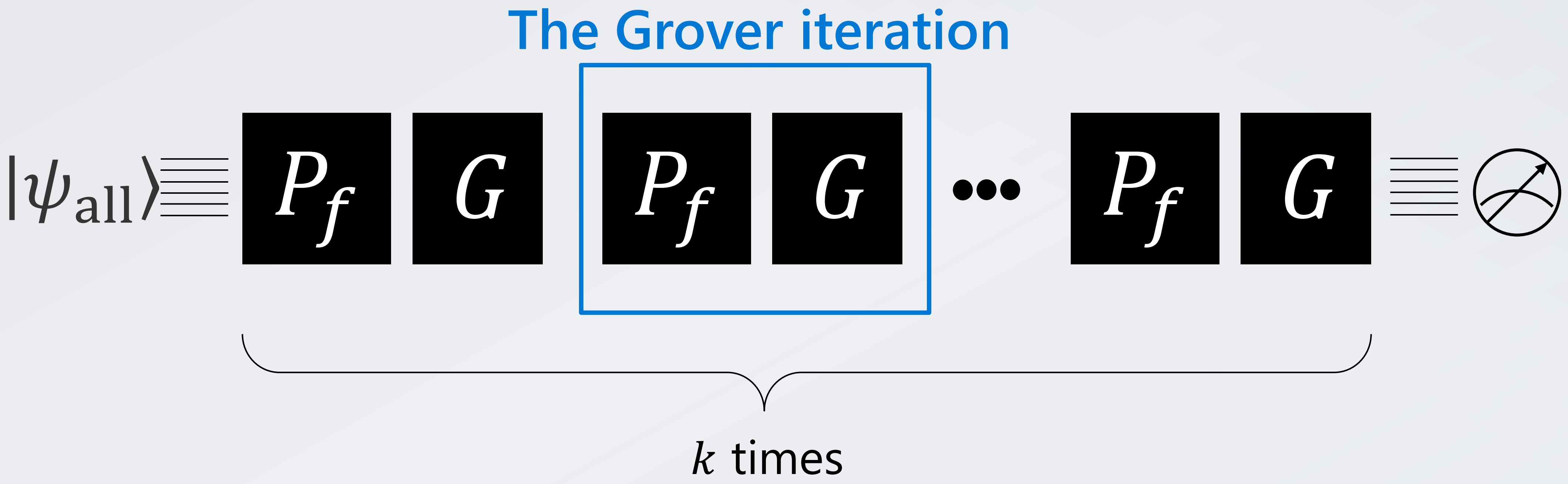
Reflection about the mean $G = 2|\psi_{\text{all}}\rangle\langle\psi_{\text{all}}| - I$ is called the Grover unitary.

Grover’s algorithm (even for the general Search problem) only uses
the oracle (P_f) and reflection about the mean (G)!

Microsoft

Grover’s search

Grover's algorithm

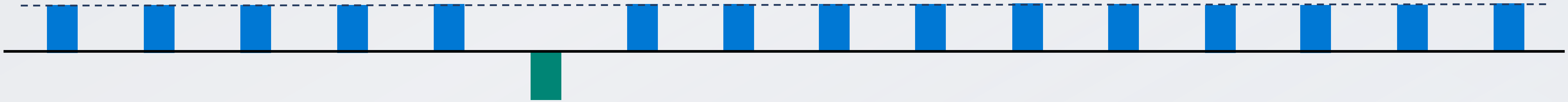


Choosing $k \approx \frac{\pi}{4}\sqrt{N}$ allows to measure the marked state x_0 (for which $f(x_0) = 1$) with high probability

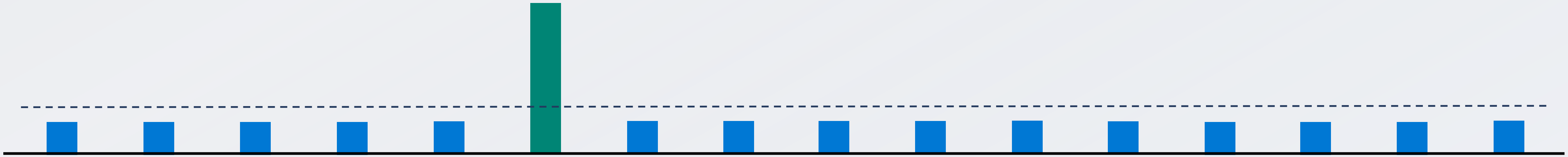
Intuition

Starting amplitudes:

After P_f:



After G:



States and reflections involved

The mean

$$|\psi_{\text{all}}\rangle = \frac{1}{\sqrt{N}} \sum_{x=1}^{N} |x\rangle$$

$$G = 2|\psi_{\text{all}}\rangle\langle\psi_{\text{all}}| - I$$

The marked item

$$|\psi_{\text{good}}\rangle = |x_0\rangle$$

$$P_f = 2\big|\psi_{\text{good}}\big\rangle\big\langle\psi_{\text{good}}\big| - I$$

The superposition of all unmarked states

$$|\psi_{\text{bad}}\rangle = \frac{1}{\sqrt{N-1}} \sum_{x:f(x)=0} |x\rangle$$

The superposition of all states

$$|\psi_{\text{all}}\rangle = \frac{1}{\sqrt{N}}|\psi_{\text{good}}\rangle + \frac{\sqrt{N-1}}{\sqrt{N}}|\psi_{\text{bad}}\rangle.$$

Visualization: definitions

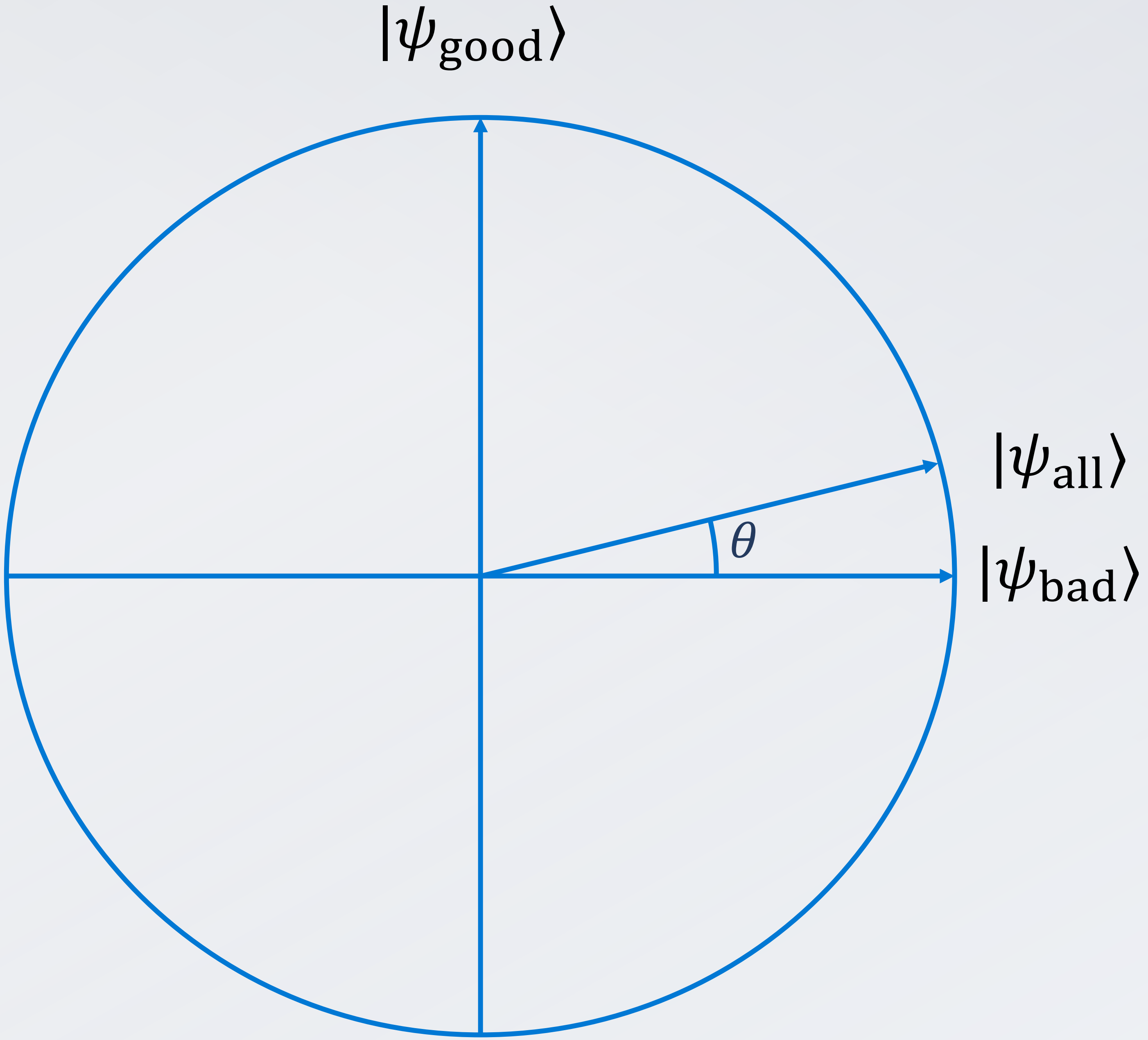
$|\psi_{\text{good}}\rangle$ = marked state

$|\psi_{\text{bad}}\rangle$ = superposition of unmarked states

$\Rightarrow \langle\psi_{\text{good}}|\psi_{\text{bad}}\rangle = 0$

$|\psi_{\text{all}}\rangle = \frac{1}{\sqrt{N}}|\psi_{\text{good}}\rangle + \sqrt{\frac{N-1}{N}}|\psi_{\text{bad}}\rangle$

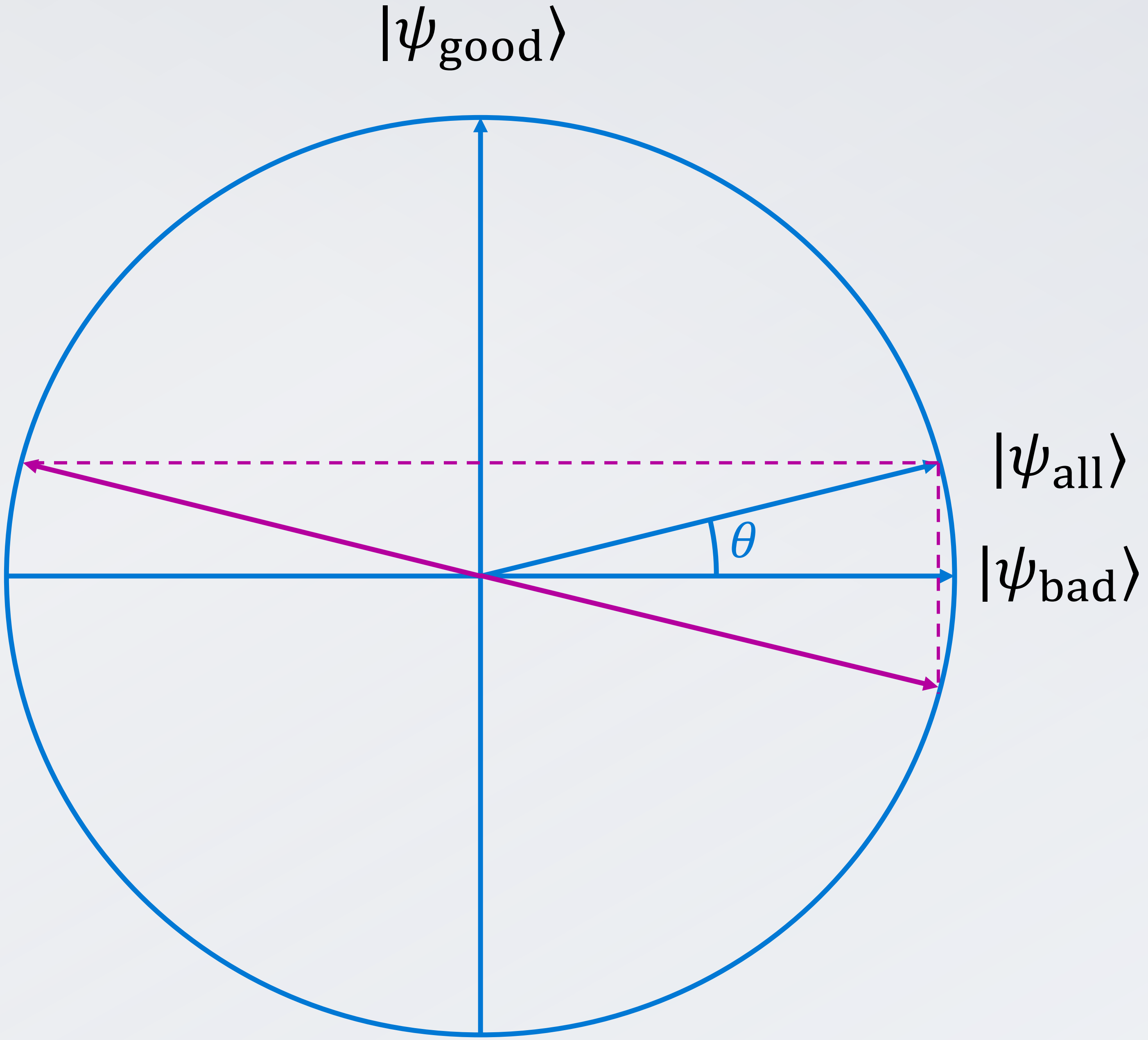
Hence $\sin\theta = \frac{1}{\sqrt{N}} \Rightarrow \theta \approx \frac{1}{\sqrt{N}}$.



Visualization: reflections

Recall that $|\psi\rangle$ and $-|\psi\rangle$ represent the same state.

Which means that reflection about $|\psi_{\text{good}}\rangle$ and reflection about $|\psi_{\text{bad}}\rangle$ do the same thing.

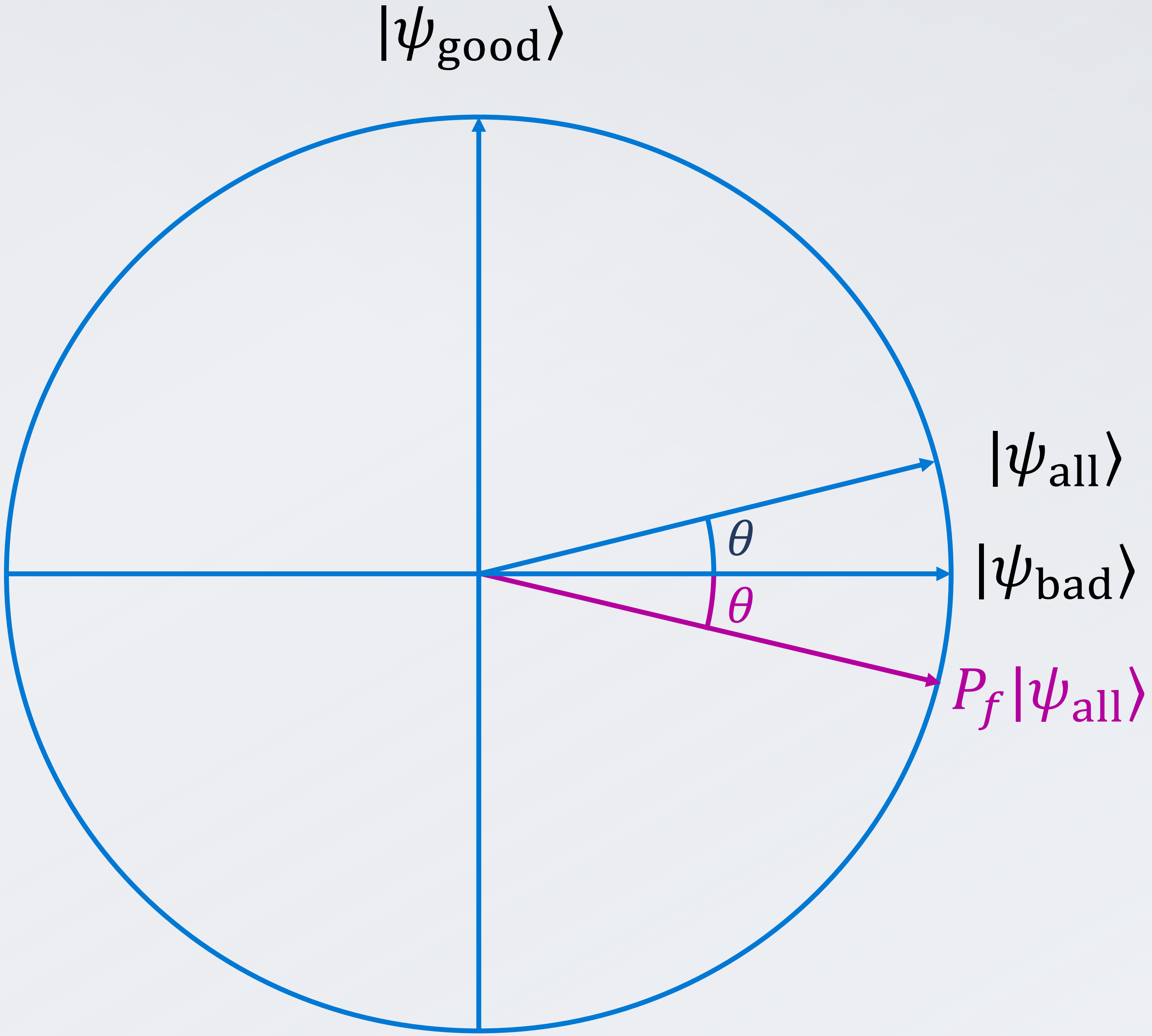


Visualization: step 1

Reflect the state of the system

about the vector $|\psi_{\text{bad}}\rangle$
(same as reflecting the state
about the vector $|\psi_{\text{good}}\rangle$)

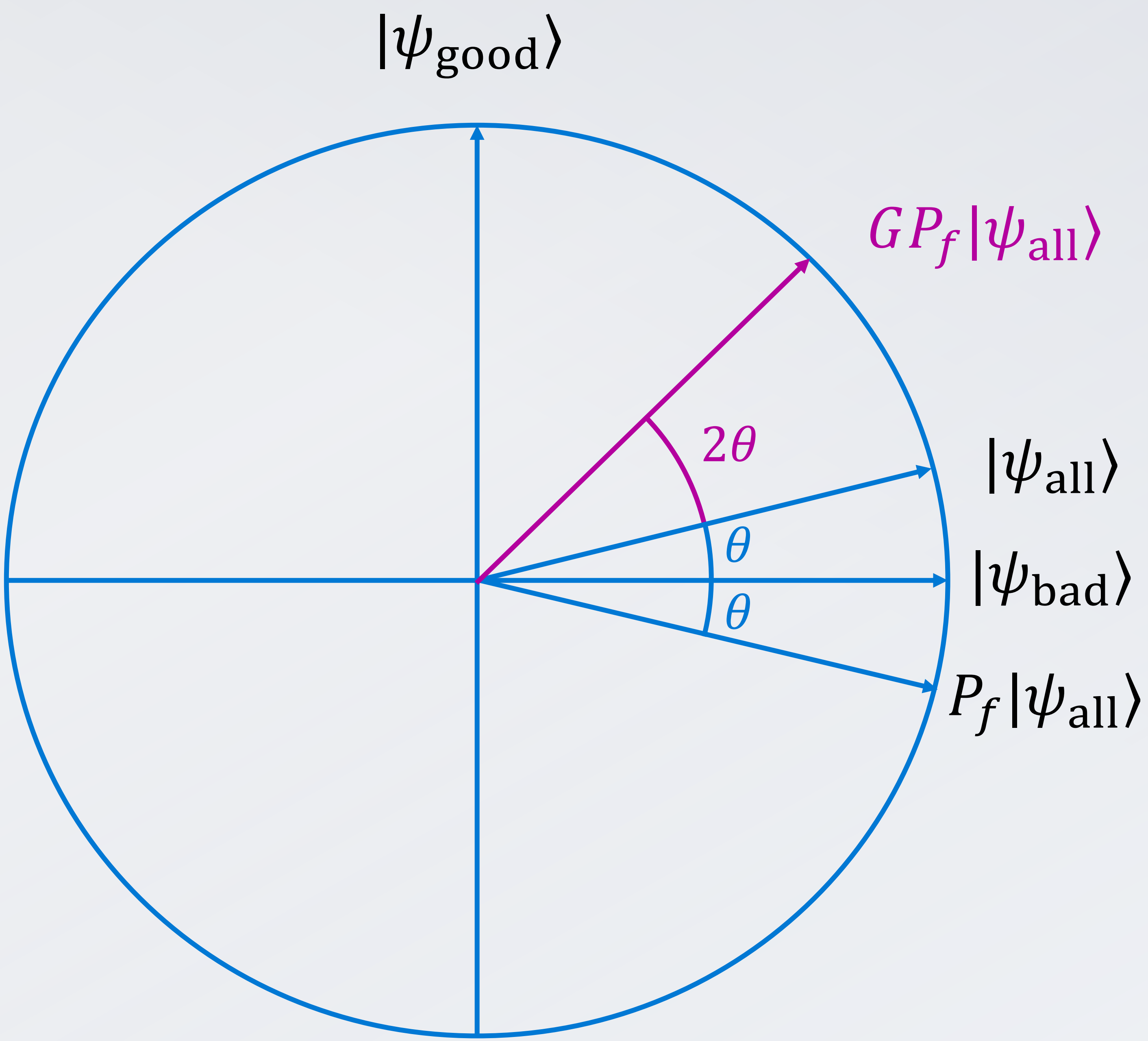
This is the step that uses
the quantum oracle
(i.e., information about
the problem we're solving)



Visualization: step 2

Reflect the state of the system across vector $|\psi_{\text{all}}\rangle$

These two reflections combined rotated the state of the system further from non-solutions $|\psi_{\text{bad}}\rangle$ and closer to $|\psi_{\text{good}}\rangle$!

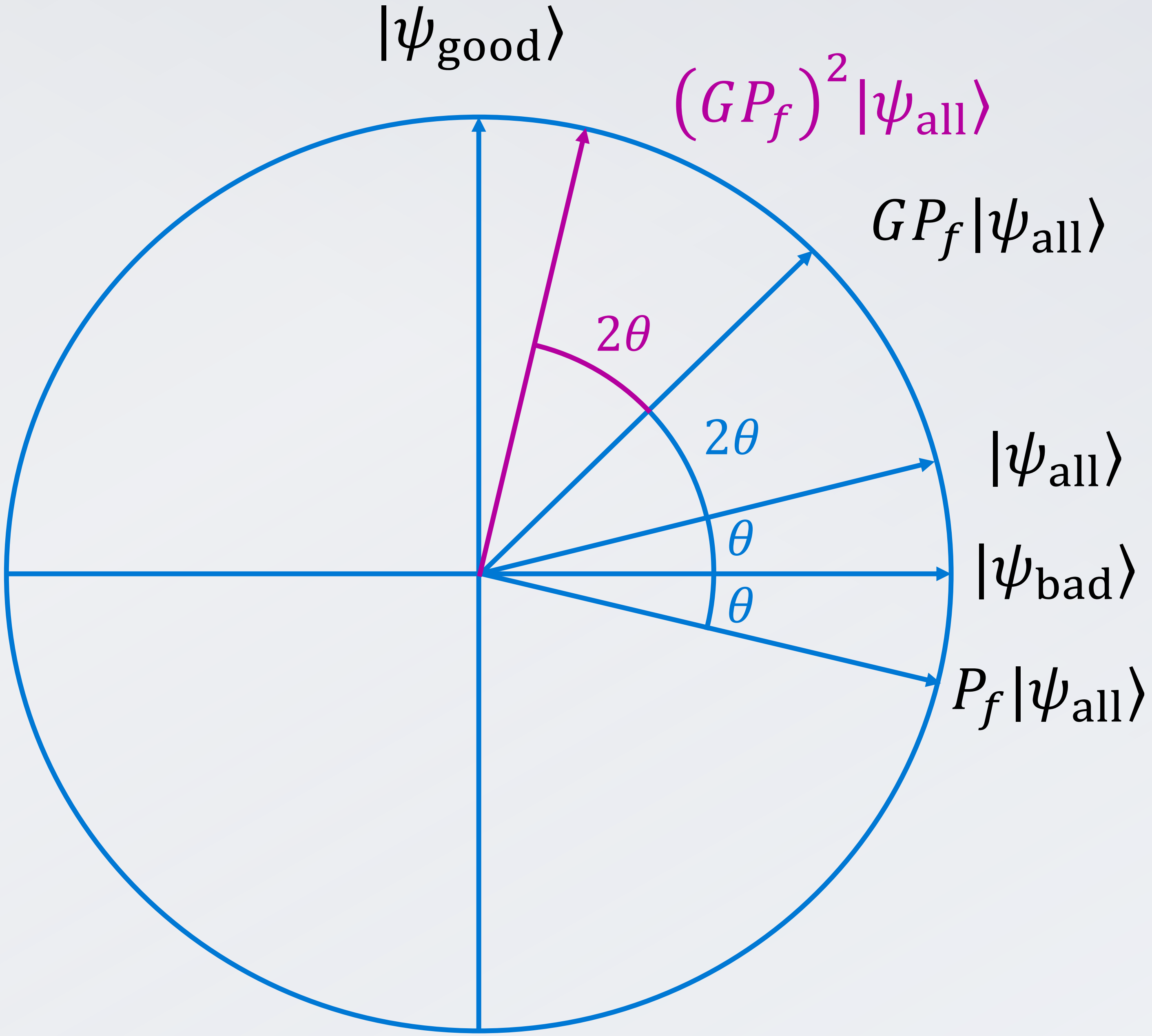


Visualizations: next steps

Product of 2 reflections = Rotation

Rotation angle = 2 × angle between the reflection axes

# iterations	angle
0	θ
1	3θ
2	5θ
k	$(2k+1)\theta$



Visualization: the goal

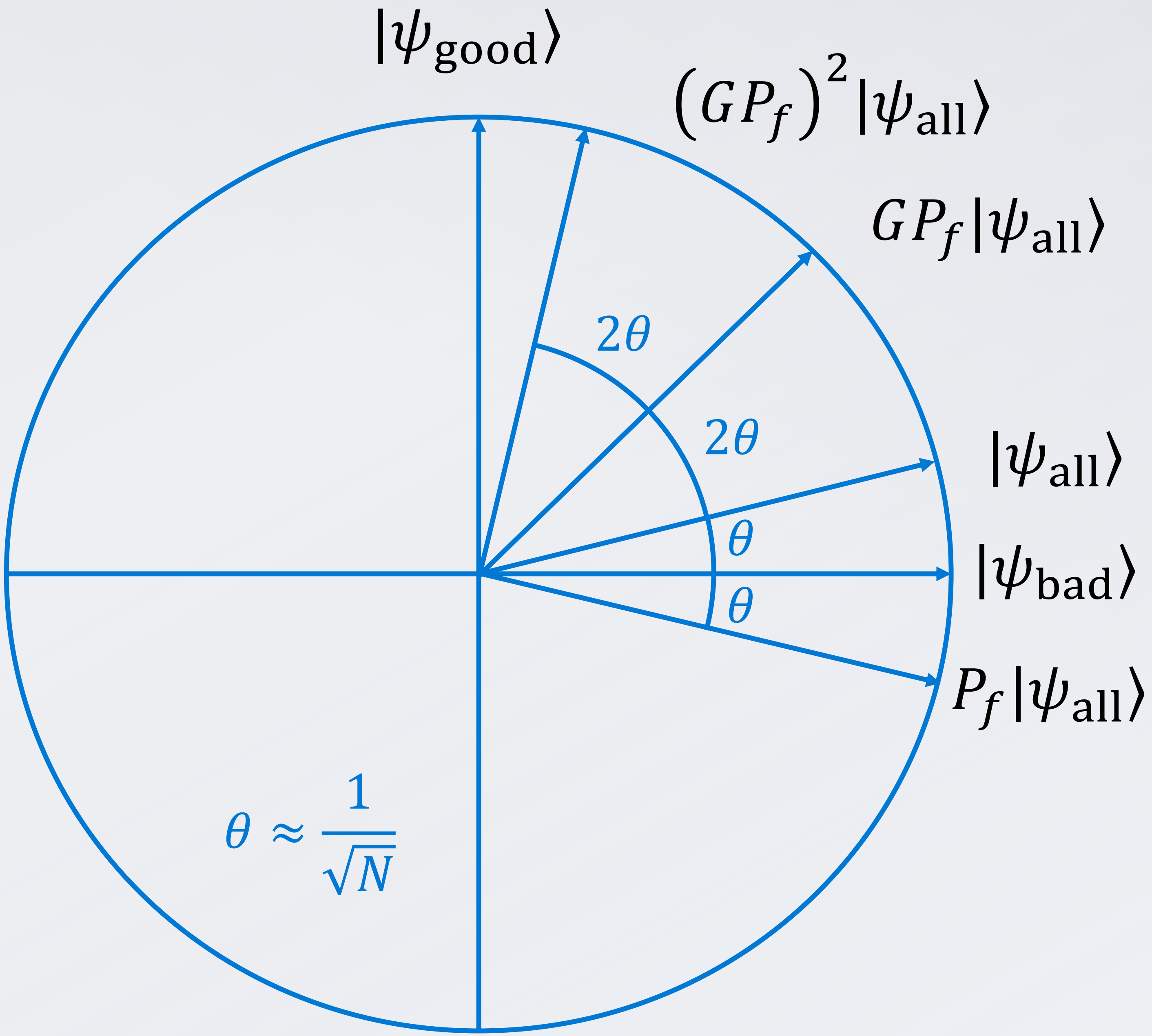
# iterations	angle
0	θ
1	3θ
2	5θ
k	$(2k+1)\theta$

We want

$$(2k+1)\theta \approx \frac{\pi}{2}$$

Choose

$$k \approx \frac{\pi}{4\theta} \approx \frac{\pi}{4}\sqrt{N}$$



Practical aspects of using Grover's search algorithm

Grover's algorithm is probabilistic

Unless $(2k+1)\theta = \frac{\pi}{2}$ and you did just the right number of iterations, there will be a non-zero failure probability

How to detect failure?

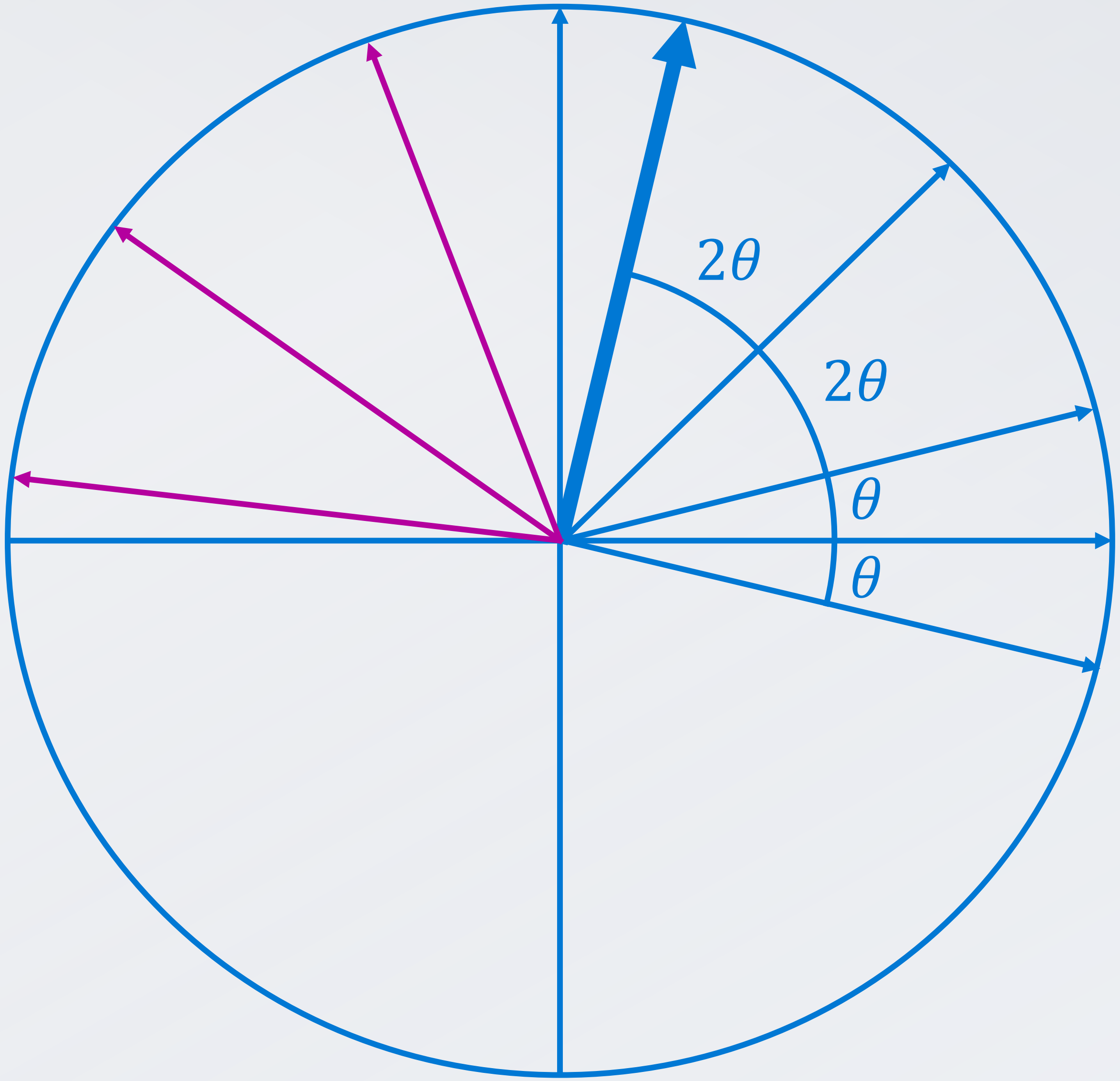
- Verify the output to check that it is indeed a solution to the problem
- Use a classical description of the problem or the same oracle to run verification

How to deal with failure?

- Re-run the algorithm from scratch

More iterations does not mean better

- There is a "sweet spot" at $k \approx \frac{\pi}{4}\sqrt{N}$
- After that doing more iterations *reduces* success probability, until we reach $k \approx \frac{2\pi}{4}\sqrt{N}$, which brings the state close to $-|\psi_{bad}\rangle$
- After that doing more iterations *increases* success probability again until we get close to $-|\psi_{\text{good}}\rangle$
- And so on



Multiple marked states

$|\psi_{\text{good}}\rangle = \frac{1}{\sqrt{M}} \sum_{x:f(x)=1} |x\rangle$

superposition of M marked states

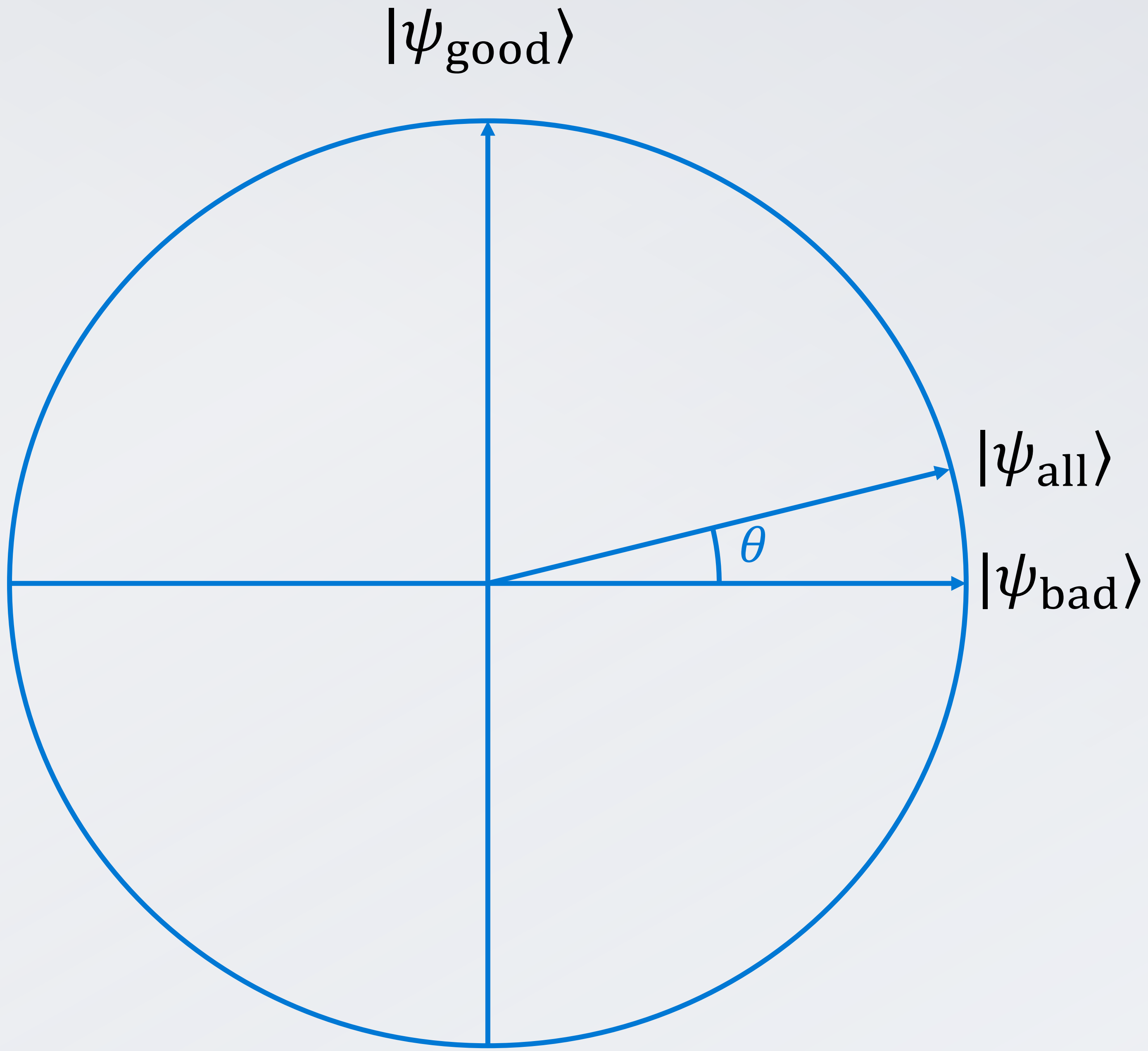
$|\psi_{\text{bad}}\rangle = \frac{1}{\sqrt{N-M}} \sum_{x:f(x)=0} |x\rangle$

superposition of $N - M$ unmarked states

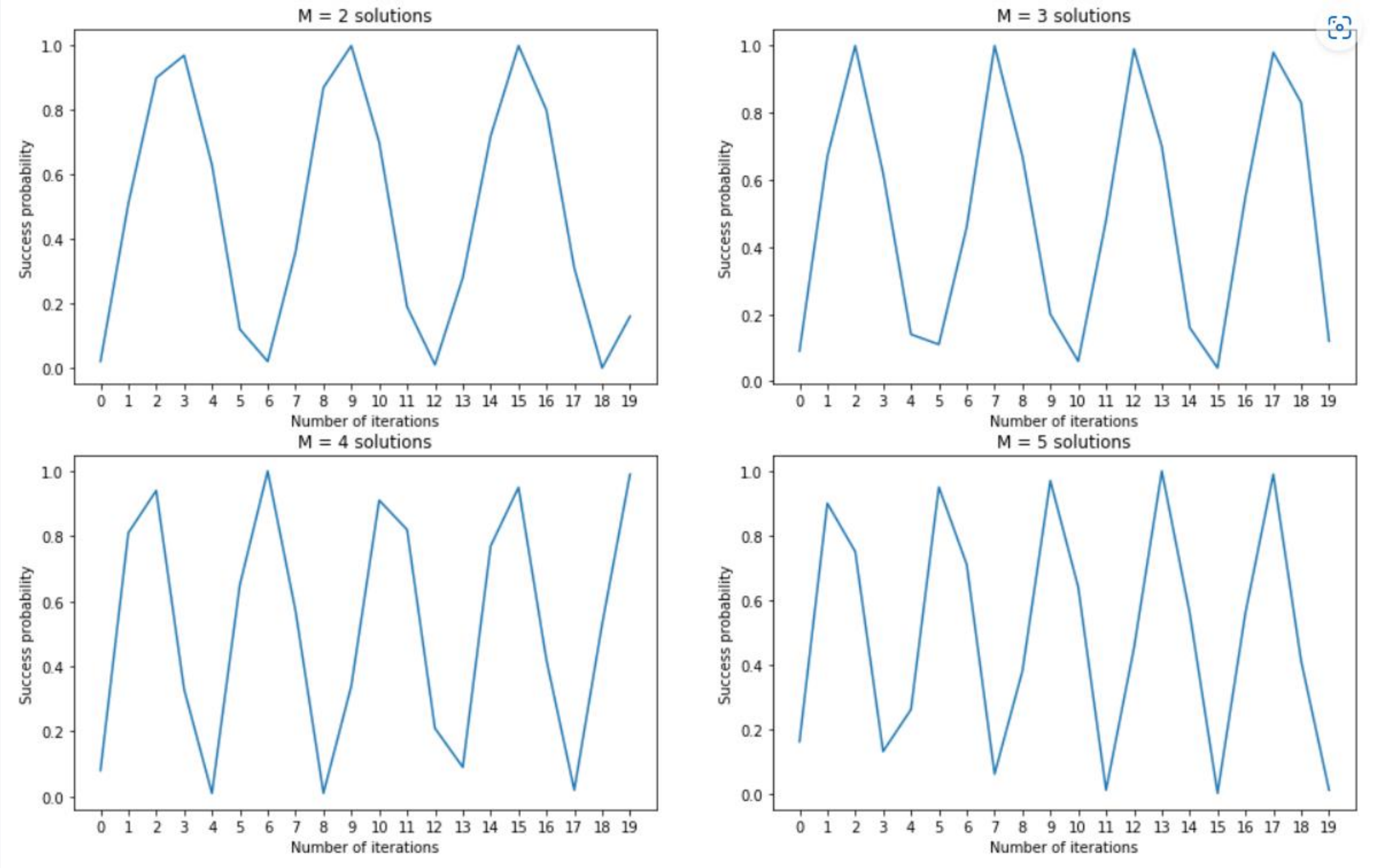
$$|\psi_{\text{all}}\rangle = \sqrt{\frac{M}{N}}\, |\psi_{\text{good}}\rangle + \sqrt{\frac{N-M}{N}}\, |\psi_{\text{bad}}\rangle$$

$$\sin\theta = \sqrt{\frac{M}{N}} \Rightarrow \theta \approx \sqrt{\frac{M}{N}}$$

$$(2k+1)\theta \approx \frac{\pi}{2} \Rightarrow k \approx \frac{\pi}{4\theta} \approx \frac{\pi}{4}\sqrt{\frac{N}{M}}$$



What if M is unknown?



Plots:
Success probability for M solutions out of 2^5 possibilities

Solution:
Choose a random $k \in \{1, \ldots \frac{\pi}{4}\sqrt{N}\}$

Is Grover's algorithm always practical?

A lot of problems can be represented as inverting a function, but…

- Grover's algorithm uses no information about problem structure; best classical algorithms exploit problem structure
- Classical algorithms can use parallel processing (easier) and benefit from getting multiple computation results at once
- Implementing the quantum oracle which encodes a problem instance on a quantum computer can be hard
- Complexity is compared in terms of queries (function evaluations); if oracle evaluation is time-consuming, advantage disappears

Not really a database search…

Why are we talking about it?

Second major quantum computing algorithm (after Shor's algorithm)

Can be used for problems which don't have efficient classical algorithms

Symmetric cryptography (AES), hash functions, etc.

Can be used to speed up other algorithms (including classical)

Given an algorithm that succeeds with probability p, we can amplify its success probability to (say) $\geq 90\%$ with $O(1/\sqrt{p})$ algorithm uses (this is known as "amplitude amplification")