OKINAWA SCHOOL IN PHYSICS: COHERENT QUANTUM DYNAMICS
Sep.28th-Oct. 9th @OIST

Interdisciplinary fields between quantum information science and physics

Keisuke Fujii
The Hakubi center for advanced research/
Graduate School of Science
Kyoto University

京都大学
KYOTO UNIVERSITY

Outline of 3 Days

Lecture 1: foundations of quantum computation

-elementary gates and universal quantum computation

-quantum algorithms

-quantum stabilizer formalism (graph state, quantum error correction)

Lecture 2: 1D quantum system

-what is quantum phase

-how useful for QIP

Lecture 3: 2D quantum system

-topologically ordered system

-how it is related to quantum error correction codes

-how topologically protected quantum computation works

Outline of 3 Days

Lecture 1: foundations of quantum computation

-elementary gates and universal quantum computation

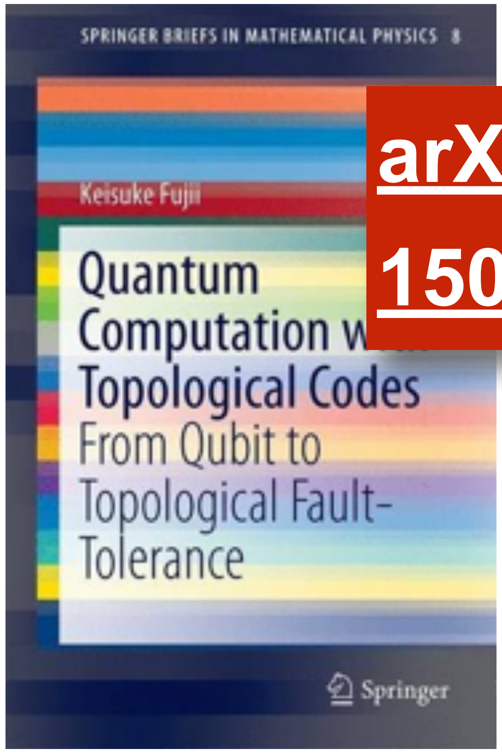
-quantum algorithms

-quantum stabilizer formalism (graph state, quantum error correction)

Lecture 2: 1D quantum system

-what is quantum phase

-how useful for QIP



arXiv: 1504.01444

Lecture 3: 2D quantum system

-topologically ordered system

-how it is related to quantum error correction codes

-how topologically protected quantum computation works

Today's Outline

Lecture 1: foundations of quantum computation

-how universal quantum computer works

elementary gates, Solovay-Kitaev algorithm

-how quantum algorithms work

Hadamard test, Kitaev's phase estimation, Shor's algo.

-how complex quantum states are described efficiently

Stabilizer formalizam, quantum error correction code, measurement-based quantum computation

Quantum bit = qubit

- Quantum bit

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\boxed{|\psi\rangle = \alpha|0\rangle + \beta|1\rangle}$$

$$\alpha, \beta \in \mathbb{C} \quad |\alpha|^2 + |\beta|^2 = 1$$

e.g. superconducting circuit, neutral atoms, ions, e-spin, n-spin

Quantum bit = qubit

◆Quantum bit

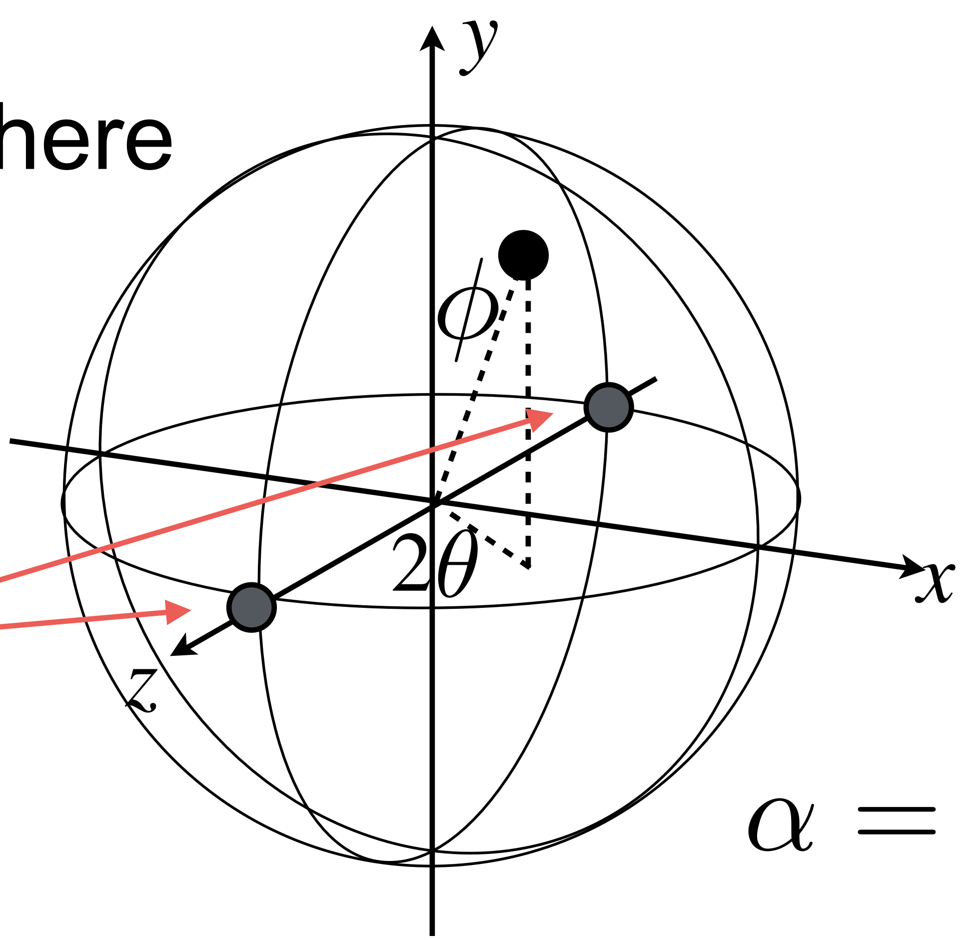
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\boxed{|\psi\rangle = \alpha|0\rangle + \beta|1\rangle}$$

$$\alpha, \beta \in \mathbb{C} \quad |\alpha|^2 + |\beta|^2 = 1$$

e.g. superconducting circuit,
neutral atoms, ions, e-spin, n-spin

Bloch sphere



classical
bit

$$\alpha = \cos\theta, \beta = e^{i\phi}\sin\theta$$

Pauli operators

- Pauli operators:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

•Anti-commute with each other, e.g. $ZX = -XZ$

Pauli operators

- Pauli operators:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

•Anti-commute with each other, e.g. $ZX = -XZ$

- Actions of the Pauli operators:

$X|0\rangle = |1\rangle \quad X|1\rangle = |0\rangle$ (bit-flip)

$Z|0\rangle = |0\rangle \quad Z|1\rangle = -|1\rangle$ (phase-flip)

$Y|0\rangle = i|1\rangle \quad Y|1\rangle = -i|0\rangle$ (bit&phase-flip + global phase)

Pauli operators

- Pauli operators:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

•Anti-commute with each other, e.g. $ZX = -XZ$

- Actions of the Pauli operators:

$X|0\rangle = |1\rangle \quad X|1\rangle = |0\rangle$ (bit-flip)

$Z|0\rangle = |0\rangle \quad Z|1\rangle = -|1\rangle$ (phase-flip)

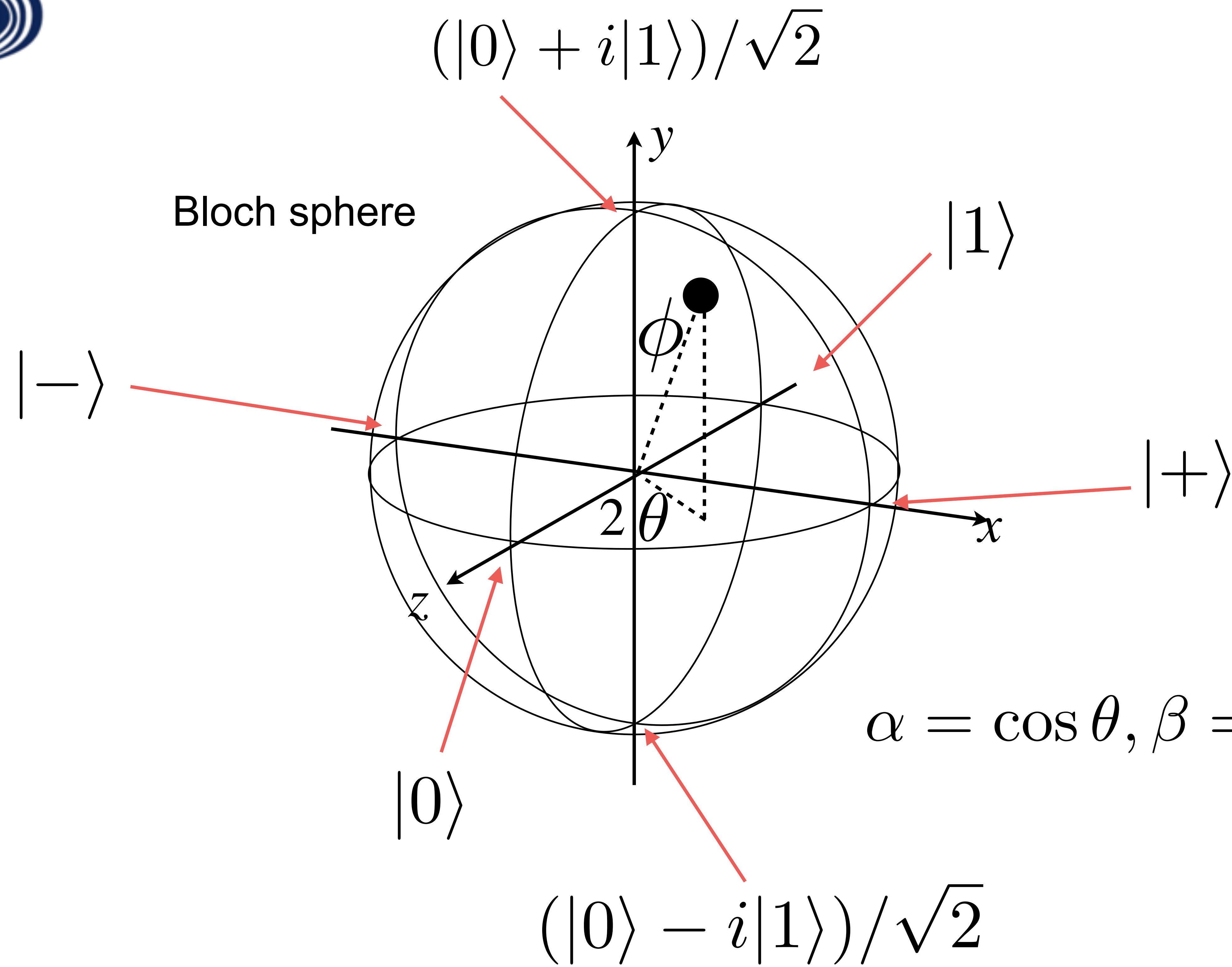
$Y|0\rangle = i|1\rangle \quad Y|1\rangle = -i|0\rangle$ (bit&phase-flip + global phase)

- Eigenstates of Pauli operators（Pauli basis）

$Z \rightarrow |0\rangle, |1\rangle$ — *Z* basis

$X \rightarrow |+\rangle \equiv (|0\rangle + |1\rangle)/\sqrt{2}, |-\rangle \equiv (|0\rangle - |1\rangle)/\sqrt{2}$ — *X* basis

Bloch sphere



$$\alpha = \cos\theta, \beta = e^{i\phi}\sin\theta$$

Clifford gates

Clifford gate maps a Pauli operator to another under conjugation:

$$A = UBU^\dagger$$

Pauli operators

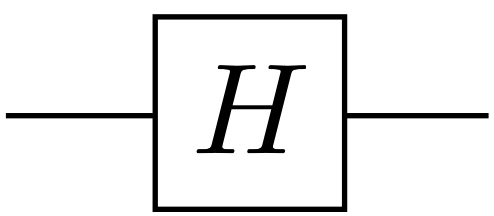
Clifford gates

Clifford gate maps a Pauli operator to another under conjugation:

$$A = UBU^\dagger$$

Pauli operators

Hadamard gate



$$= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\boxed{HXH = Z}$$

$$|+\rangle = H|0\rangle, \quad |-\rangle = H|1\rangle$$

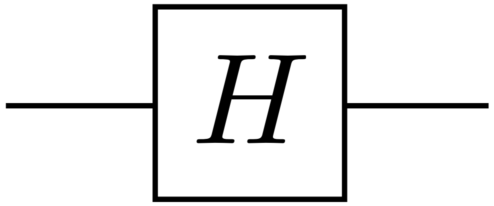
Clifford gates

Clifford gate maps a Pauli operator to another under conjugation:

$$A = UBU^\dagger$$

Pauli operators

Hadamard gate

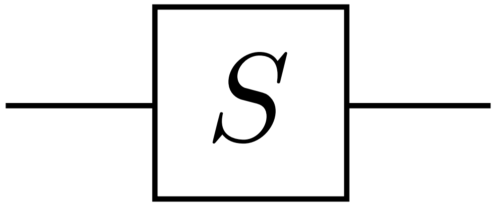


$$= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\boxed{HXH = Z}$$

$$|+\rangle = H|0\rangle, \quad |-\rangle = H|1\rangle$$

Phase gate



$$= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

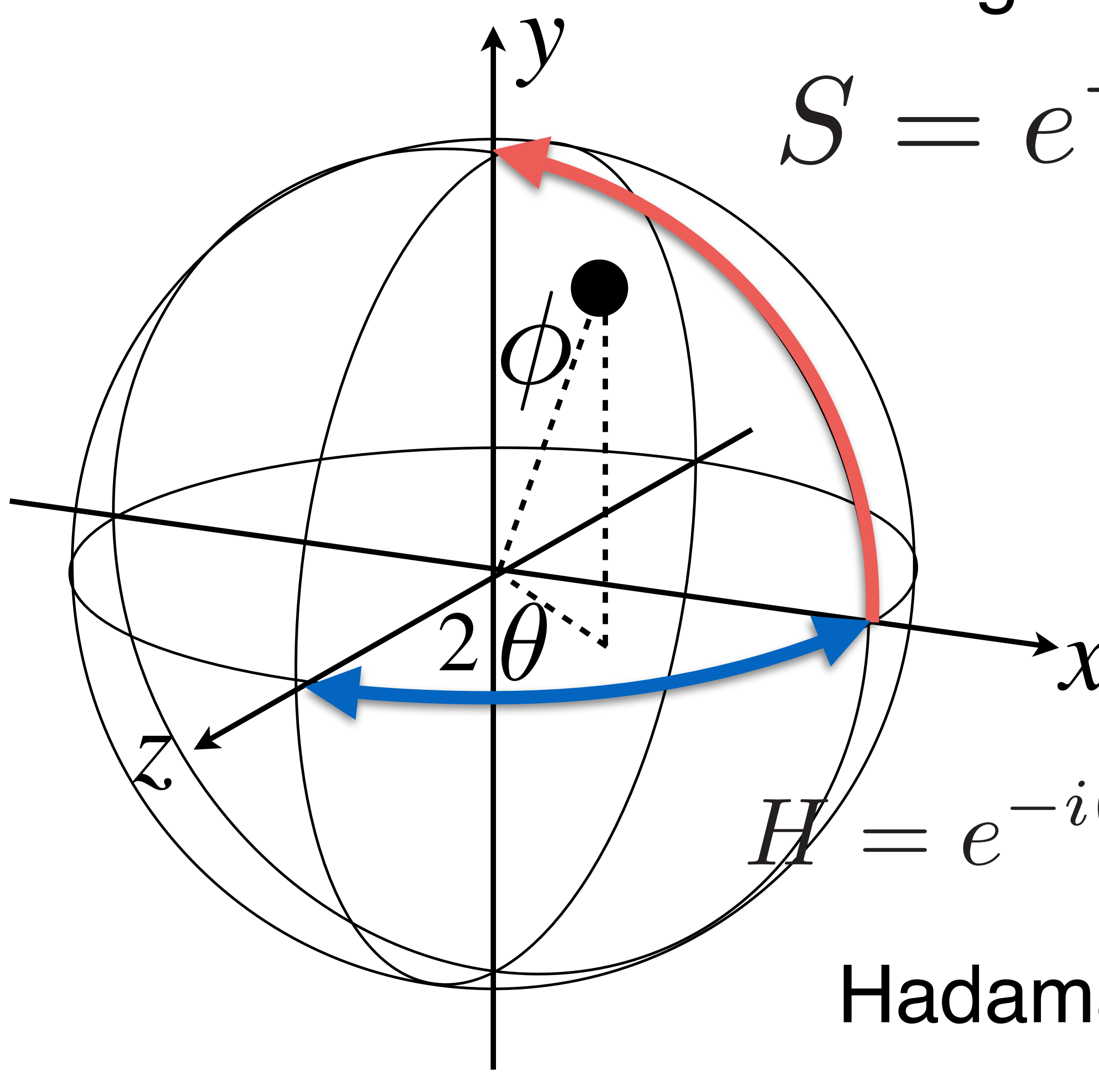
$$\boxed{SXS^\dagger = Y}$$

$$S|0\rangle = |0\rangle, \quad S|1\rangle = i|1\rangle$$

Clifford gates on Bloch sphere

Phase gate

$$S = e^{-i(\pi/4)Z}$$



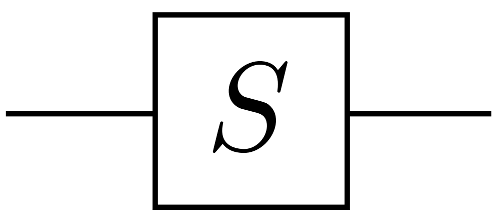
$$H = e^{-i(\pi/4)Y} Z$$

Hadamard gate

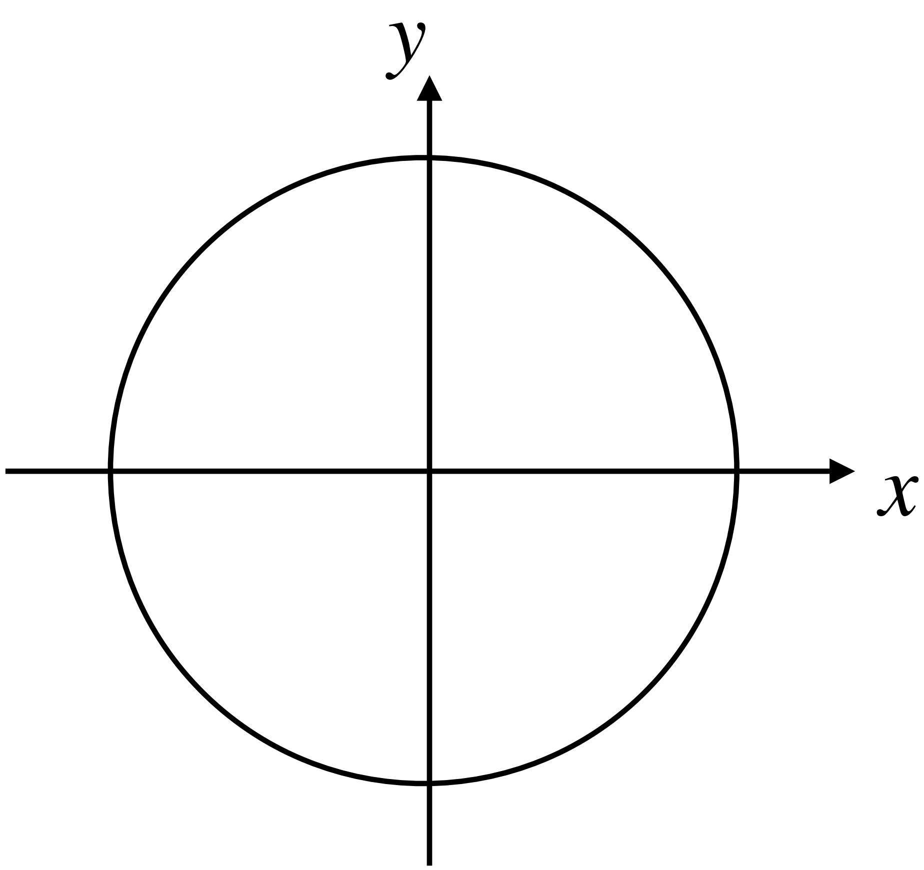
Clifford gates are not universal

$\pi/4$ gate cannot realize an arbitrary rotation:

Phase gate



$$S = e^{-i(\pi/4)Z}$$

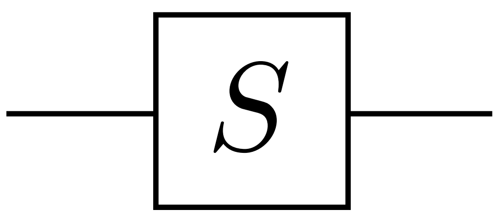


We need a rotation with an irrational angle to π.

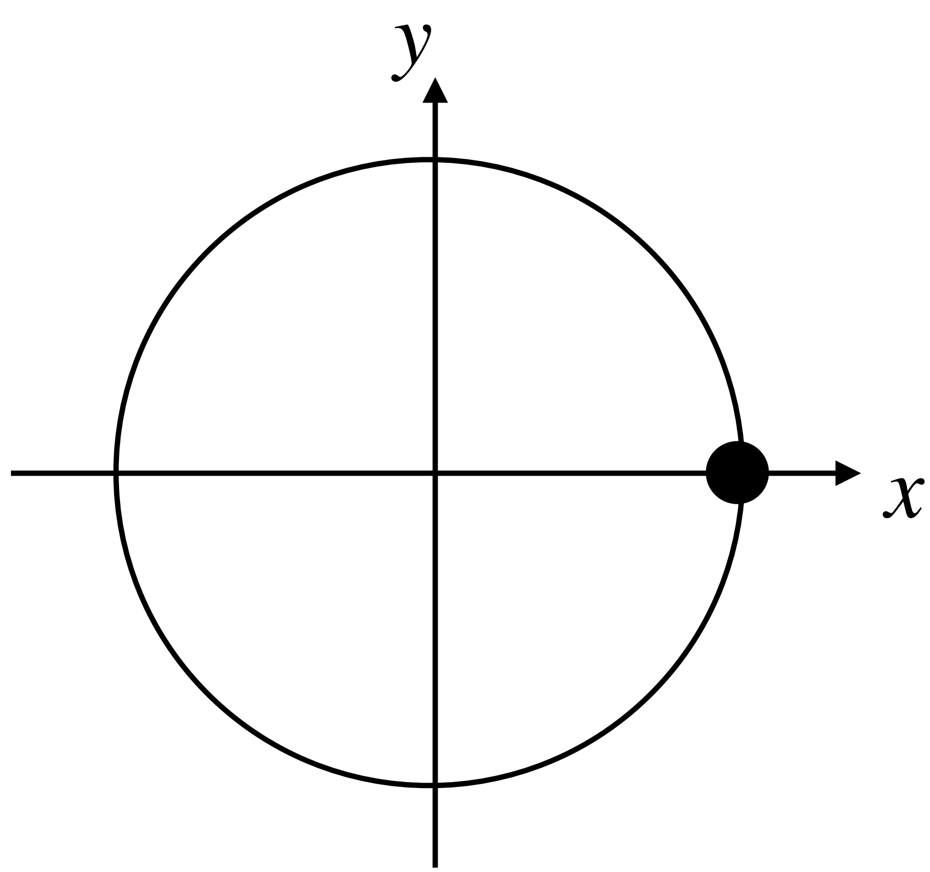
Clifford gates are not universal

$\pi/4$ gate cannot realize an arbitrary rotation:

Phase gate



$$S = e^{-i(\pi/4)Z}$$

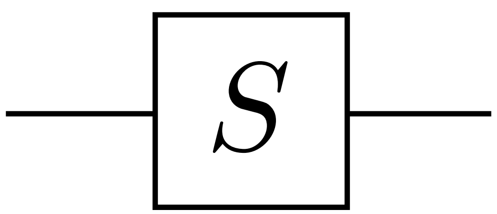


We need a rotation with an irrational angle to π.

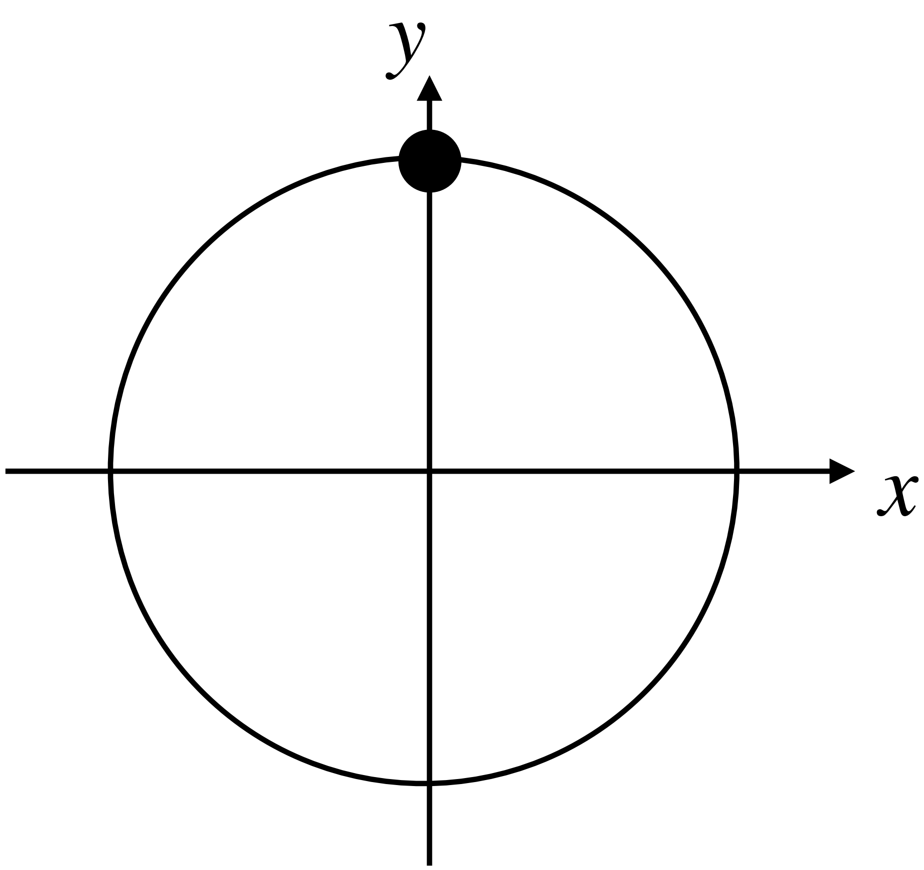
Clifford gates are not universal

$\pi/4$ gate cannot realize an arbitrary rotation:

Phase gate



$$S = e^{-i(\pi/4)Z}$$

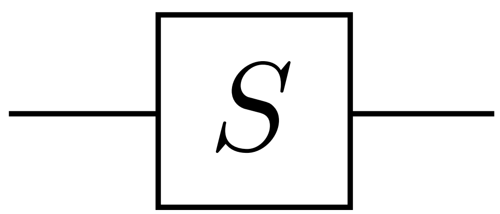


We need a rotation with an irrational angle to π.

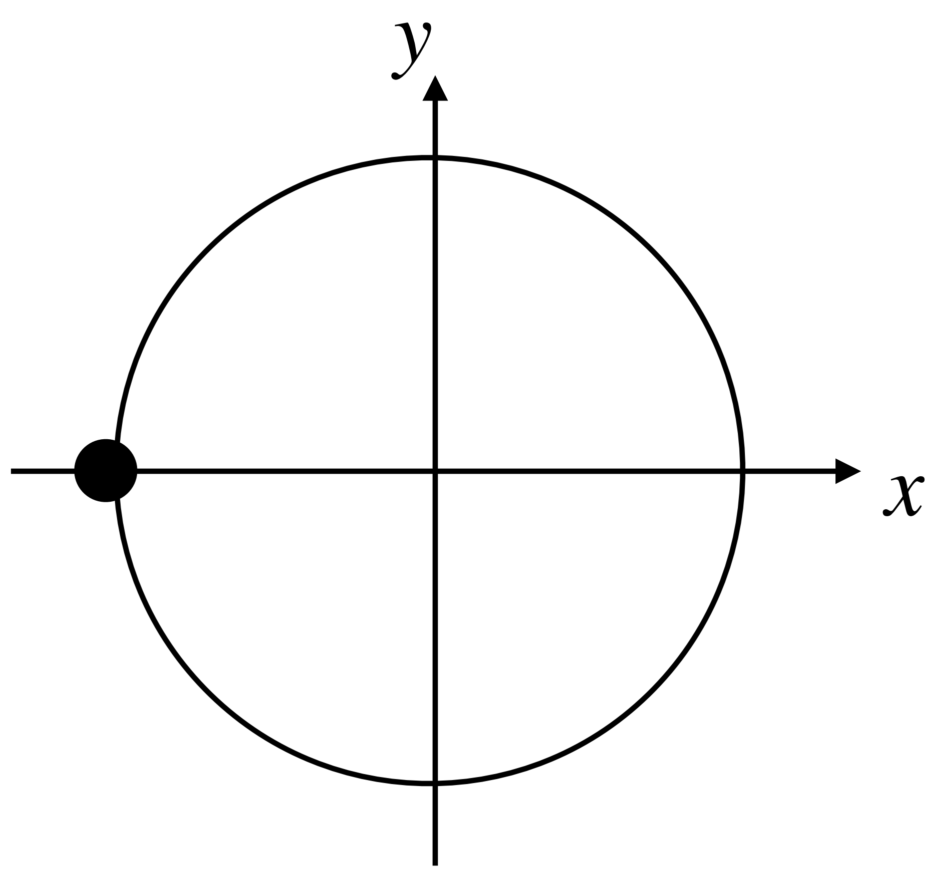
Clifford gates are not universal

$\pi/4$ gate cannot realize an arbitrary rotation:

Phase gate



$$S = e^{-i(\pi/4)Z}$$



We need a rotation with an irrational angle to π.

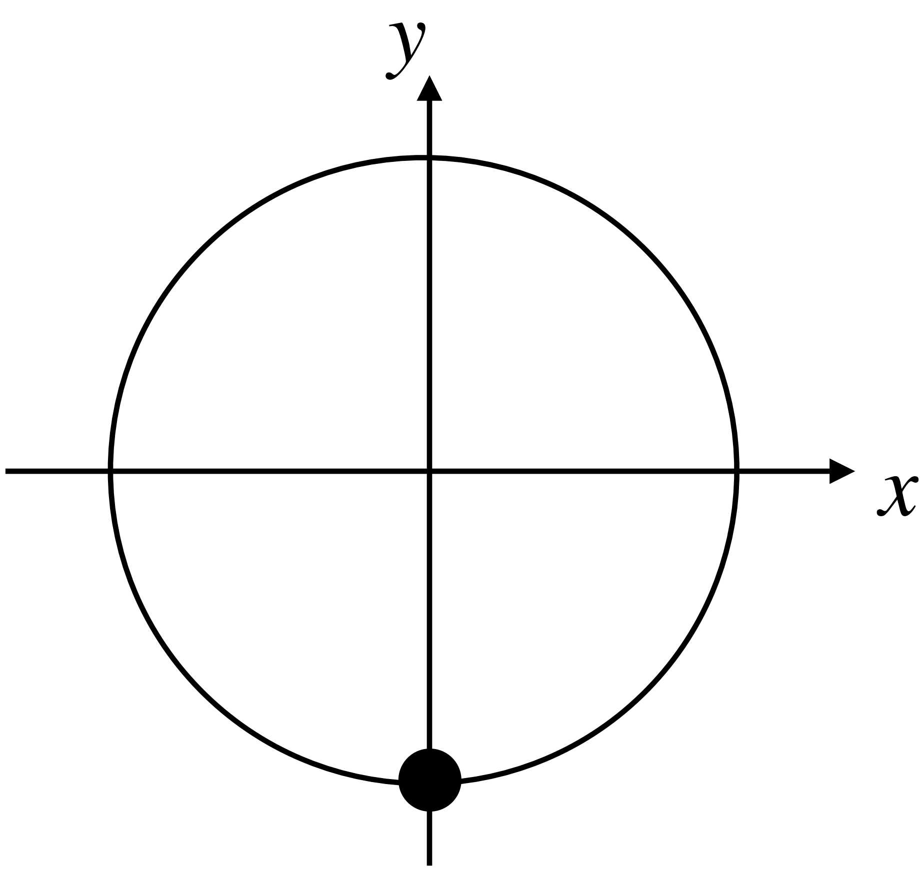
Clifford gates are not universal

$\pi/4$ gate cannot realize an arbitrary rotation:

Phase gate

$$
-\boxed{S}-
$$

$$S = e^{-i(\pi/4)Z}$$

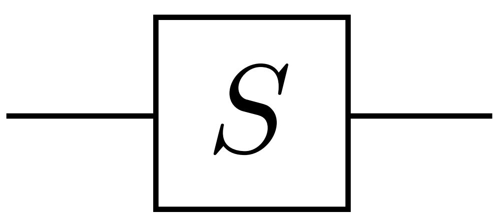


We need a rotation with an irrational angle to π.

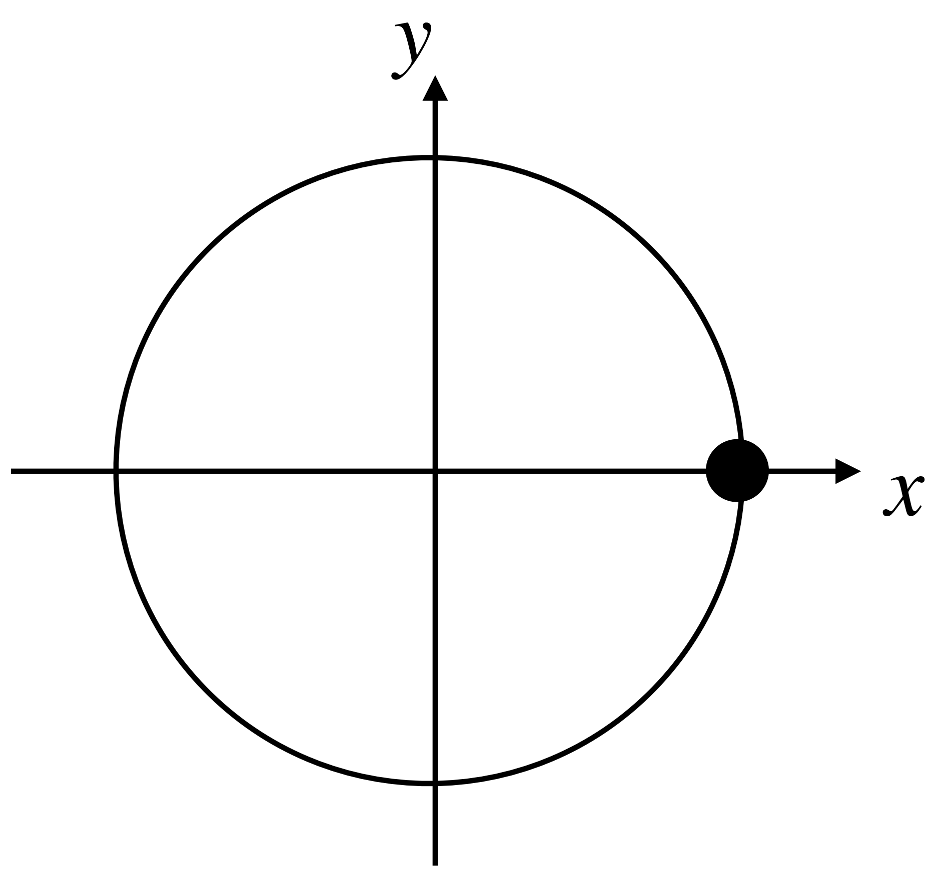
Clifford gates are not universal

$\pi/4$ gate cannot realize an arbitrary rotation:

Phase gate



$$S = e^{-i(\pi/4)Z}$$

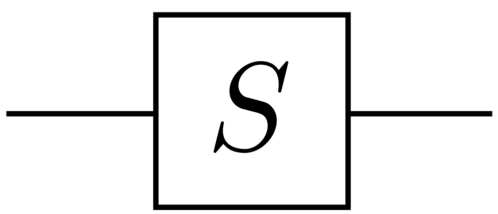


We need a rotation with an irrational angle to π.

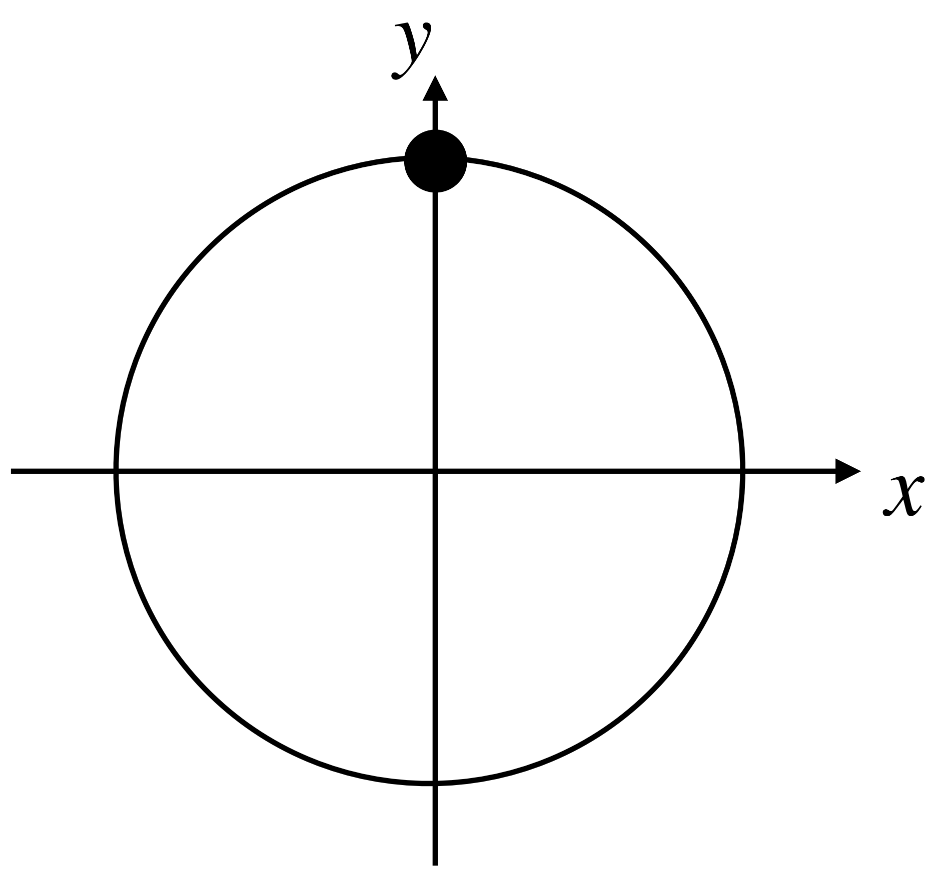
Clifford gates are not universal

$\pi/4$ gate cannot realize an arbitrary rotation:

Phase gate



$$S = e^{-i(\pi/4)Z}$$

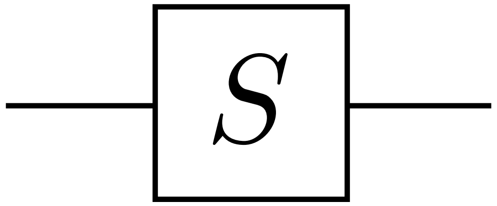


We need a rotation with an irrational angle to π.

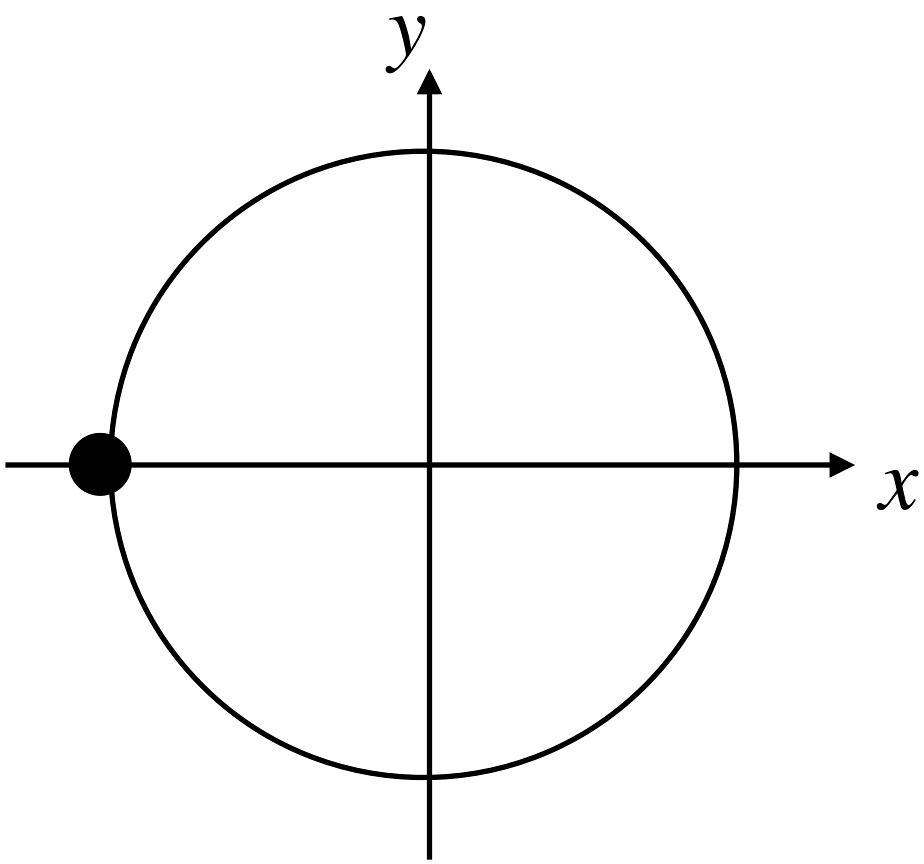
Clifford gates are not universal

π/4 gate cannot realize an arbitrary rotation:

Phase gate



$$S = e^{-i(\pi/4)Z}$$

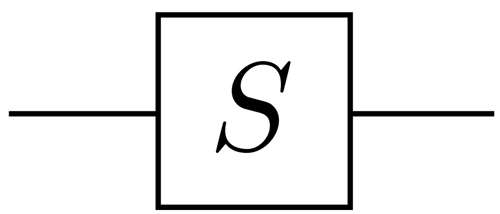


We need a rotation with an irrational angle to π.

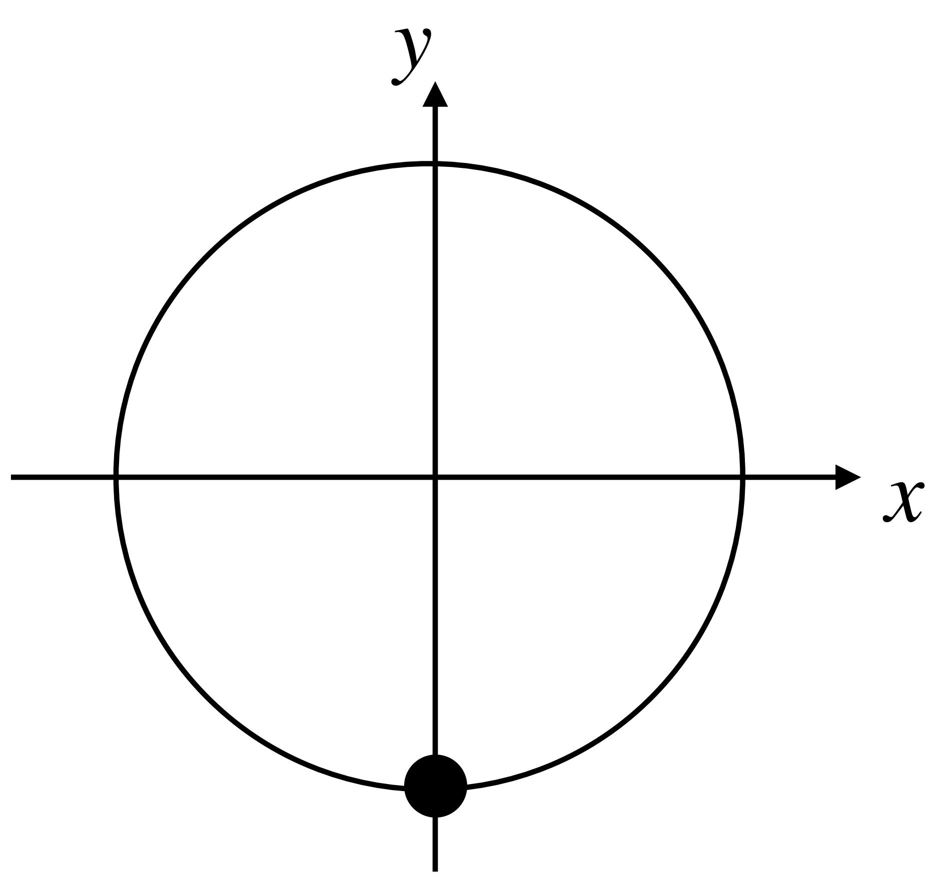
Clifford gates are not universal

$\pi/4$ gate cannot realize an arbitrary rotation:

Phase gate



$$S = e^{-i(\pi/4)Z}$$

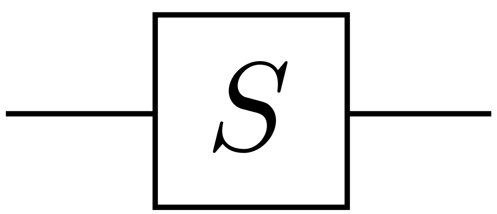


We need a rotation with an irrational angle to π.

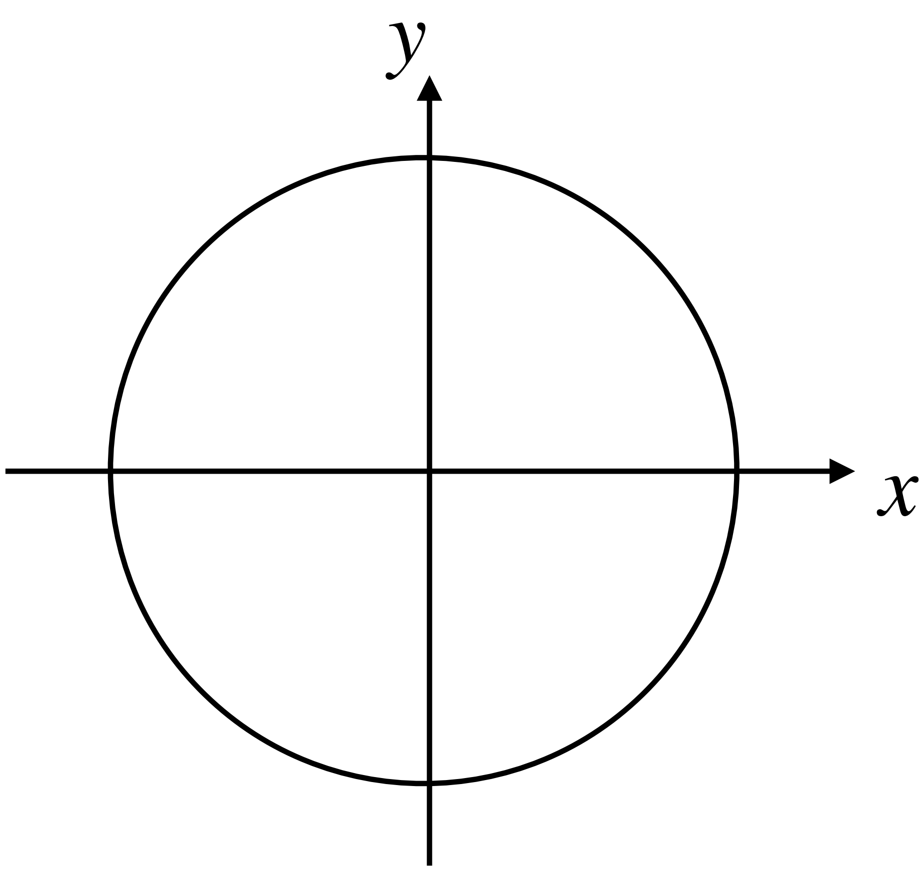
Clifford gates are not universal

$\pi/4$ gate cannot realize an arbitrary rotation:

Phase gate

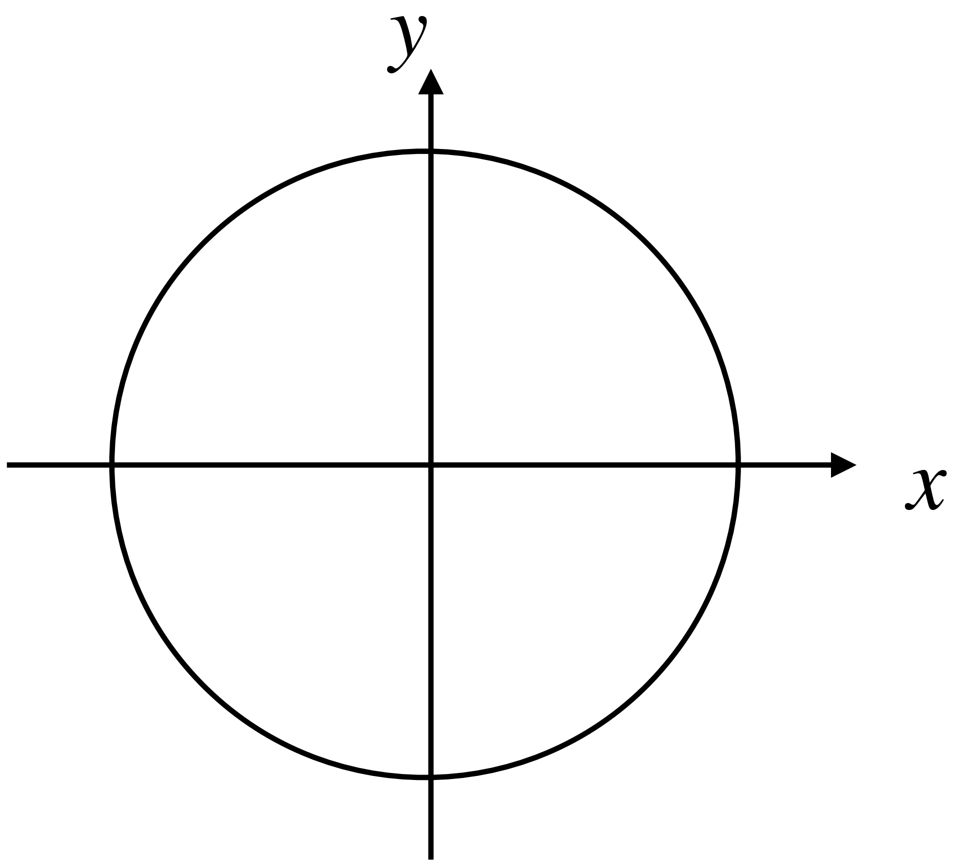


$$S = e^{-i(\pi/4)Z}$$



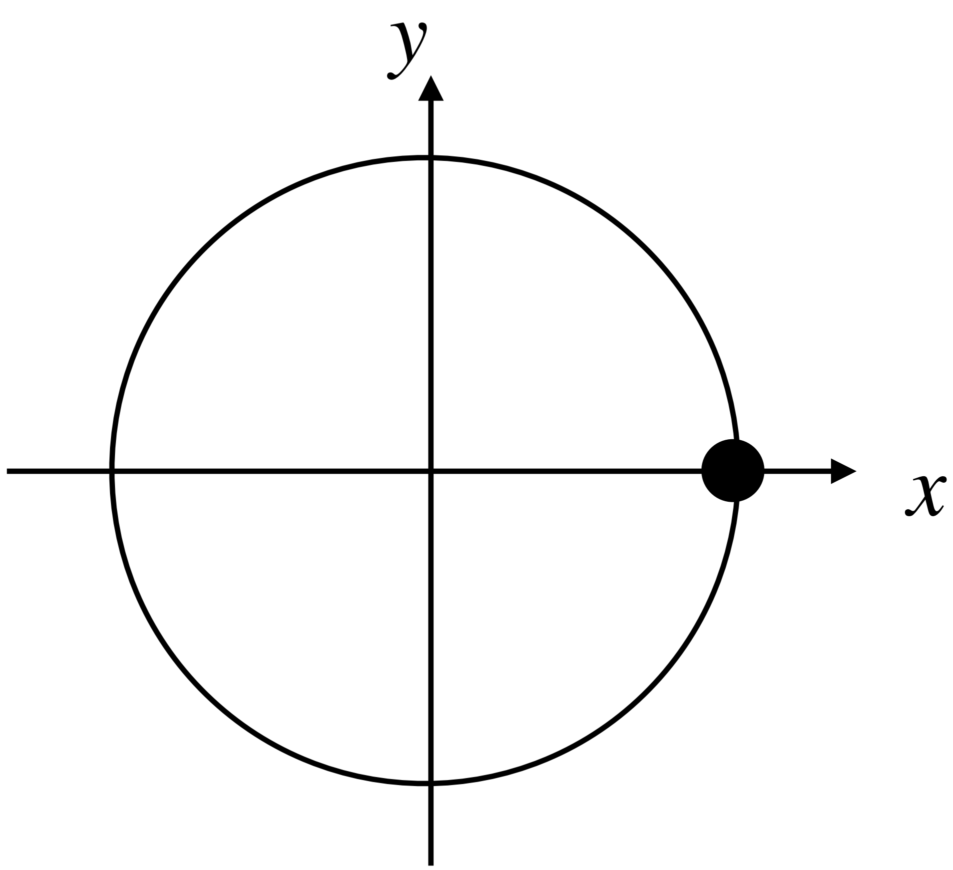
We need a rotation with an irrational angle to π.

Density implies rapid cover



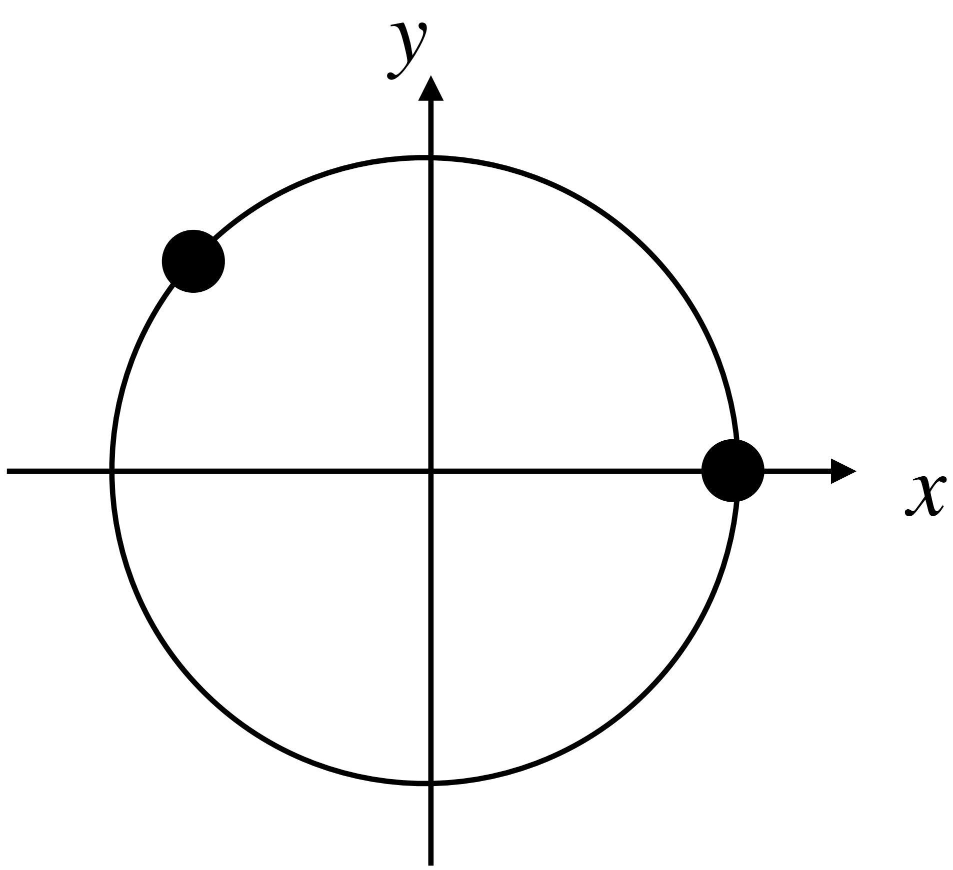
$$(3-\sqrt{5})\pi$$

Density implies rapid cover



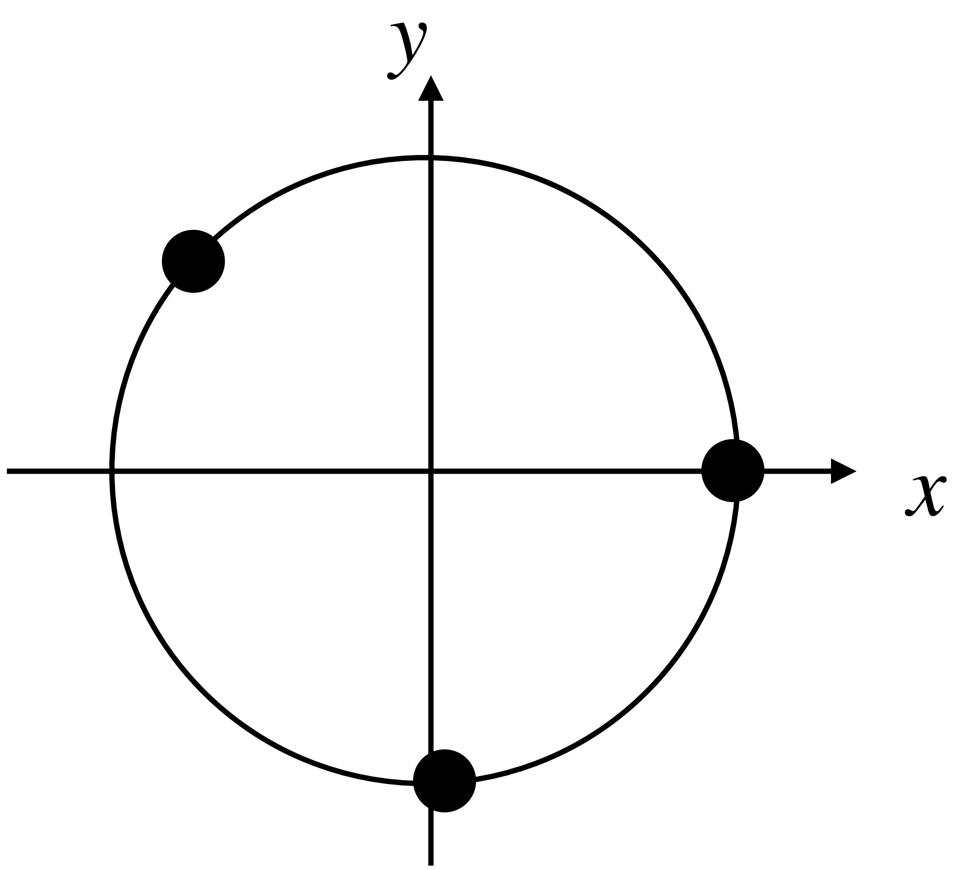
$$(3-\sqrt{5})\pi$$

Density implies rapid cover



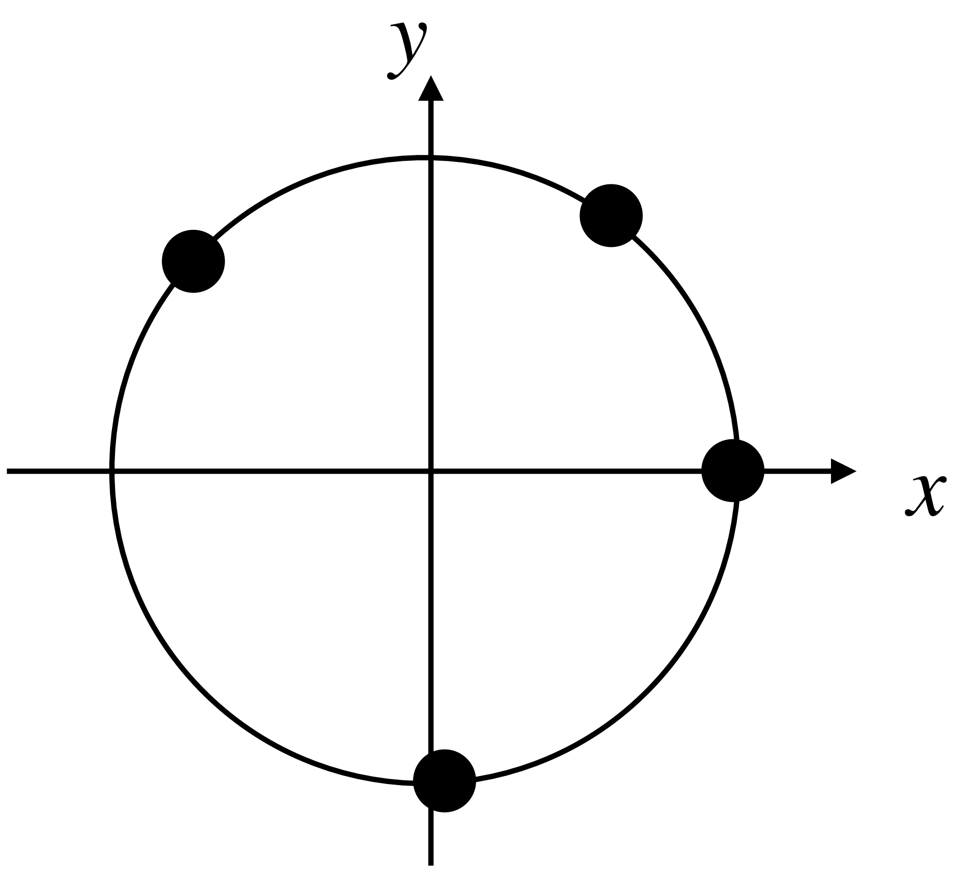
$$(3-\sqrt{5})\pi$$

Density implies rapid cover



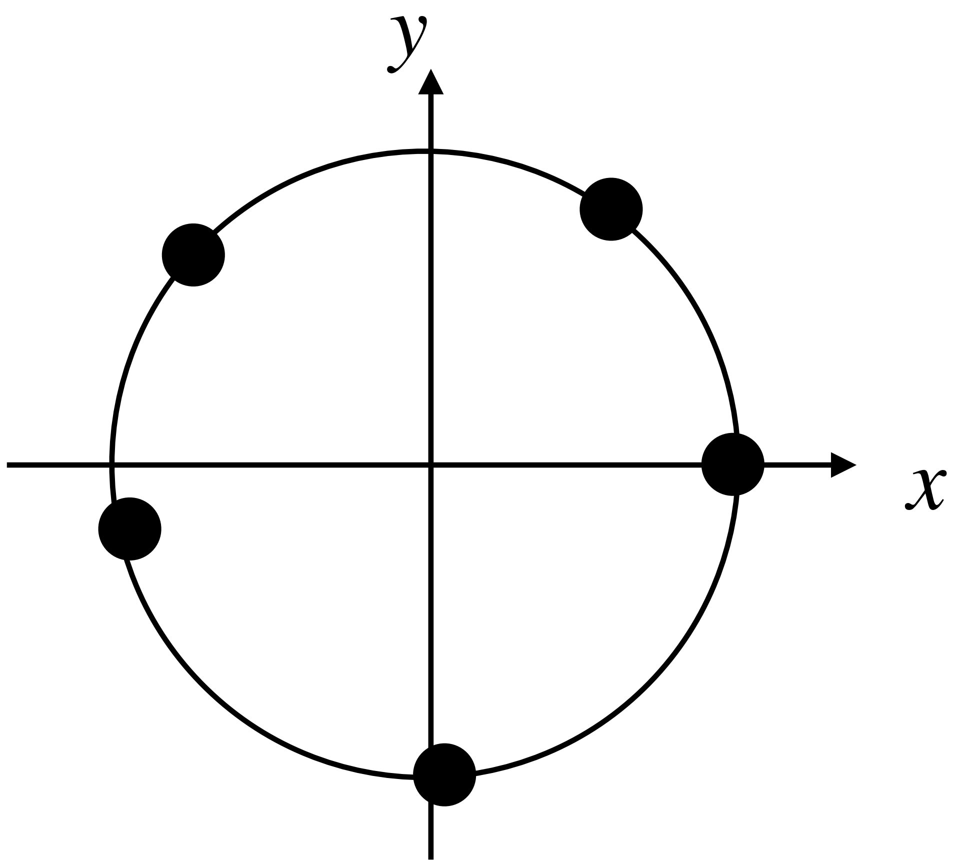
$(3-\sqrt{5})\pi$

Density implies rapid cover



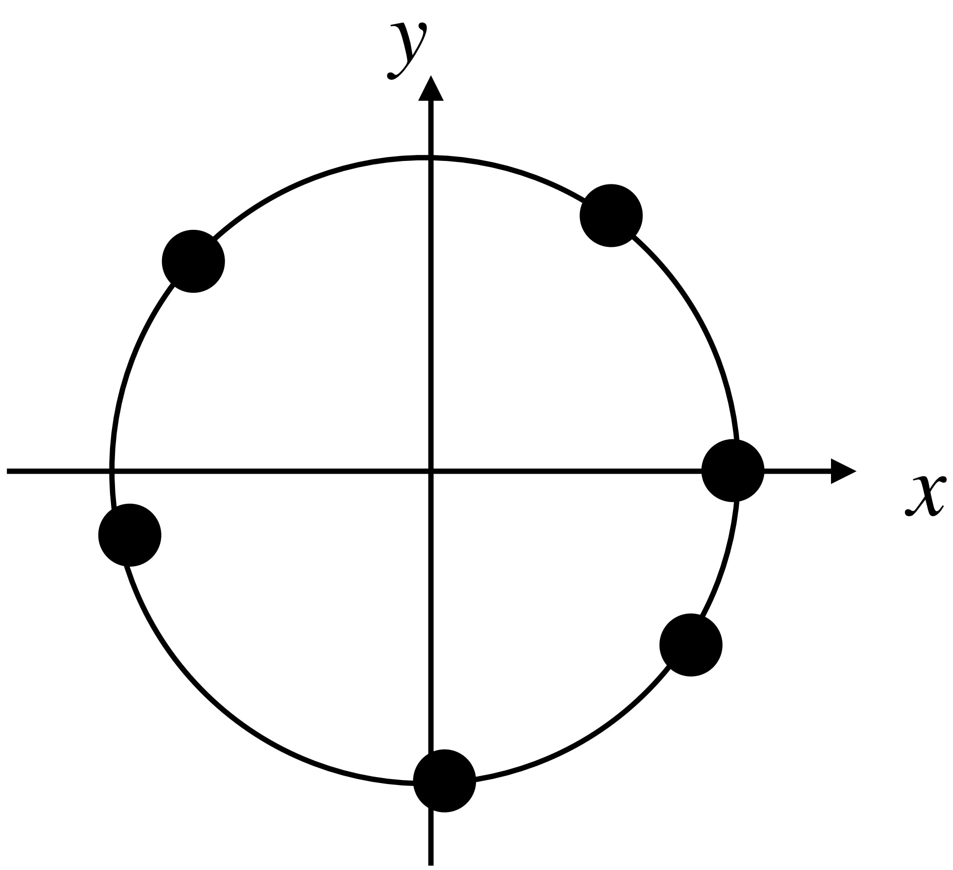
$(3-\sqrt{5})\pi$

Density implies rapid cover



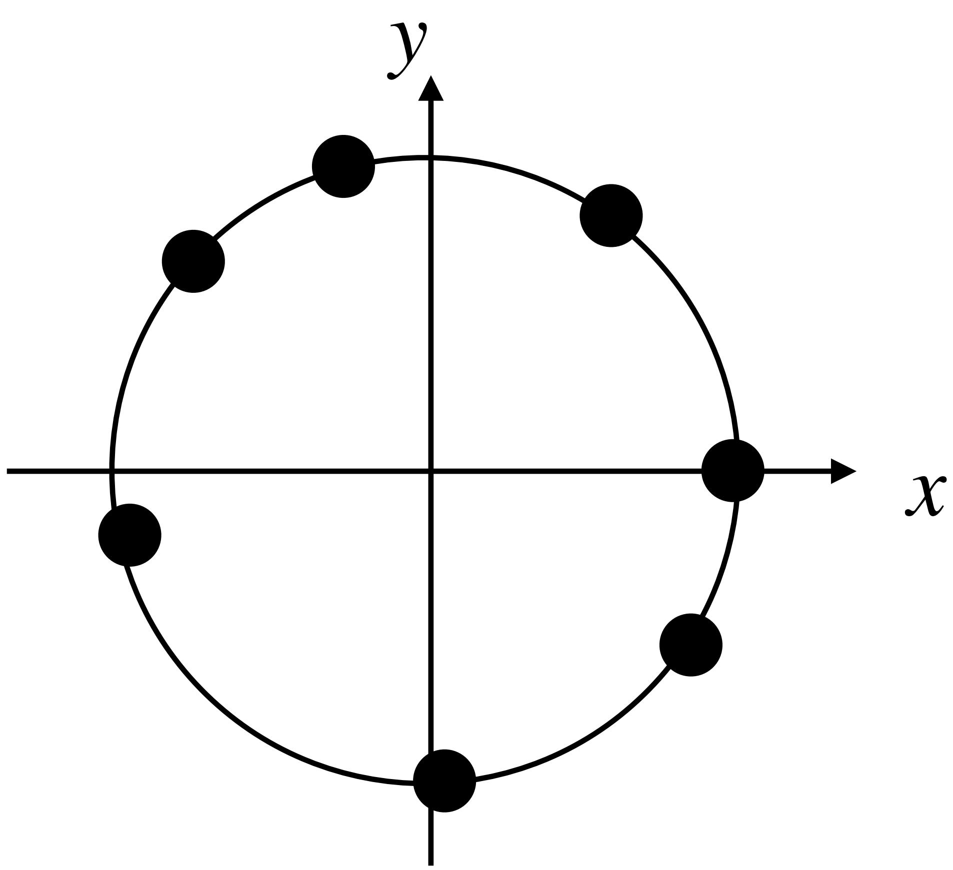
$(3-\sqrt{5})\pi$

Density implies rapid cover



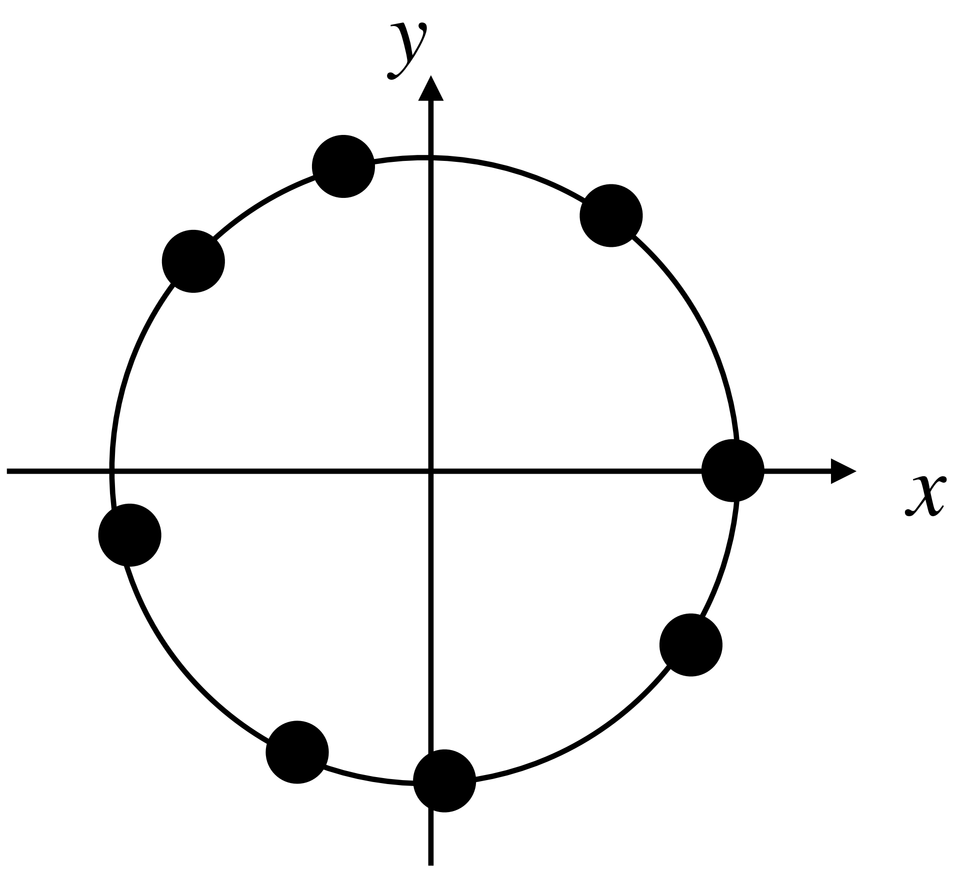
$$(3-\sqrt{5})\pi$$

Density implies rapid cover



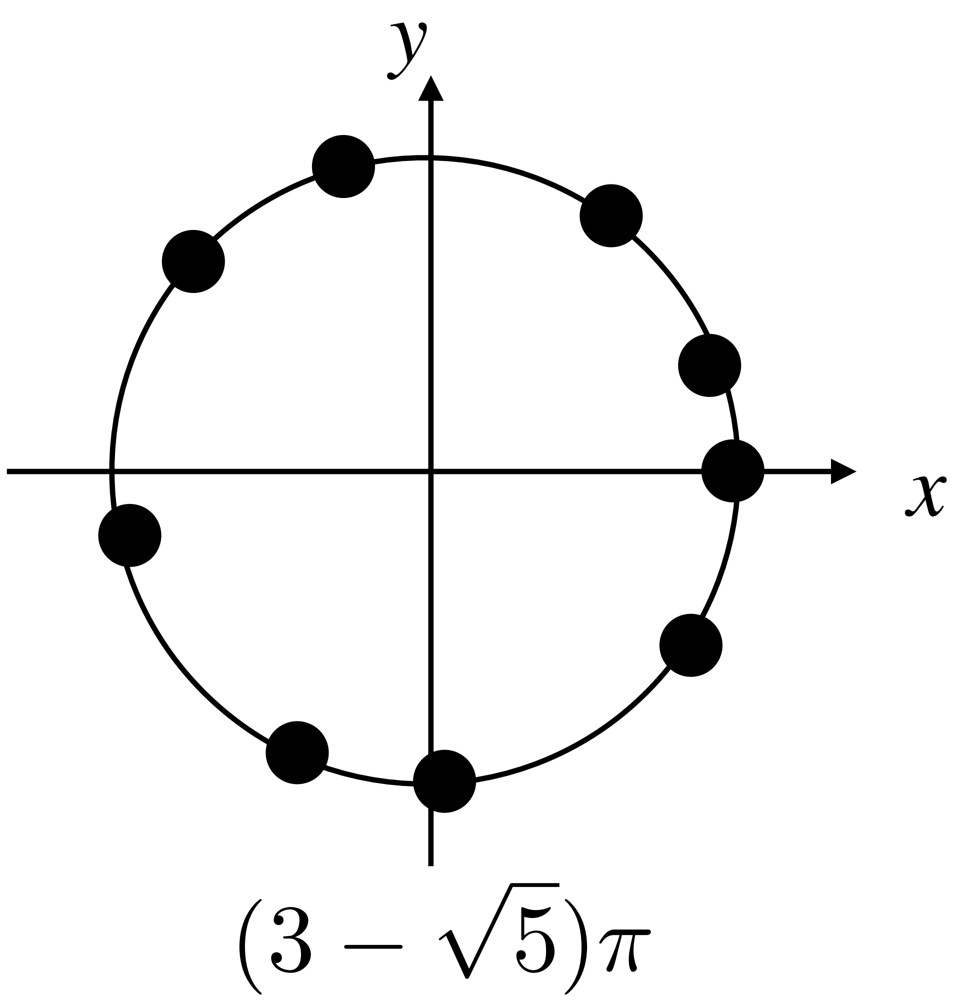
$(3-\sqrt{5})\pi$

Density implies rapid cover

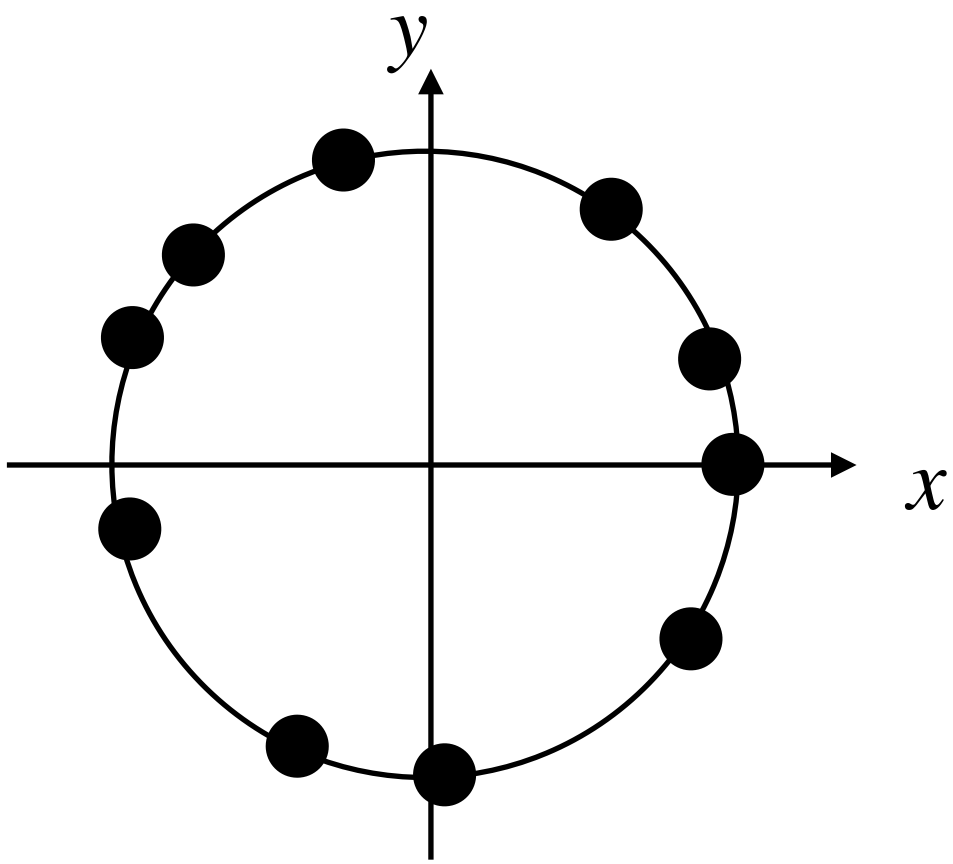


$$(3-\sqrt{5})\pi$$

Density implies rapid cover

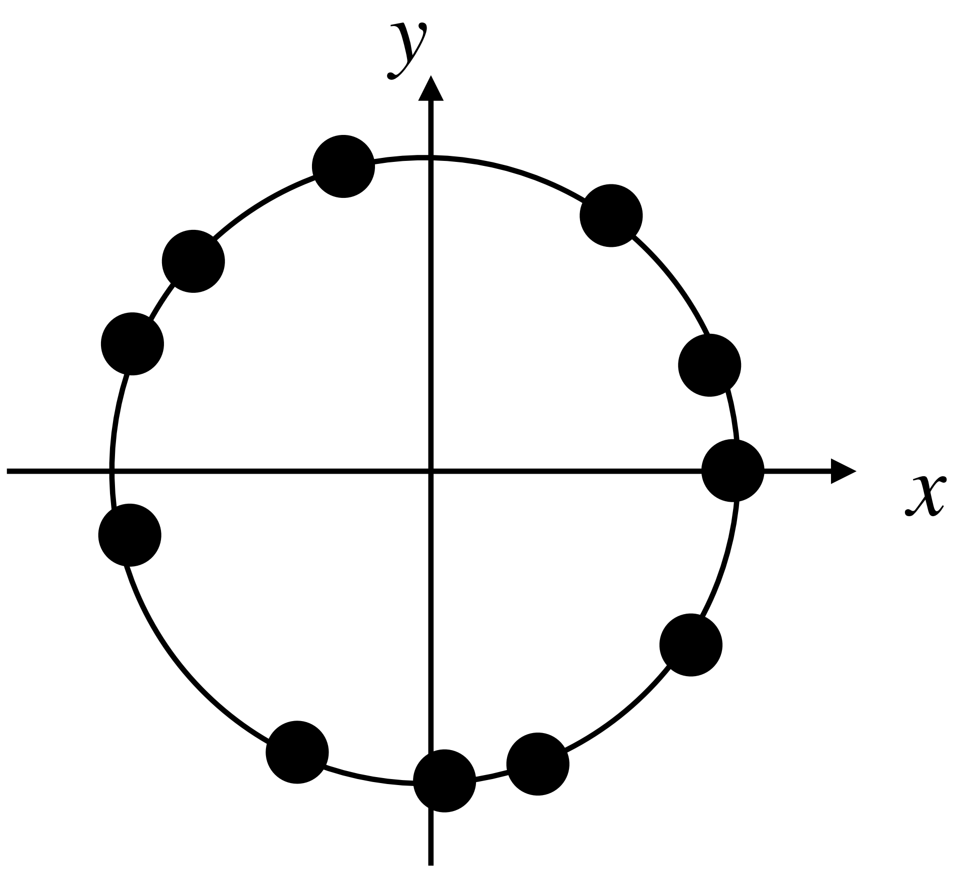


Density implies rapid cover



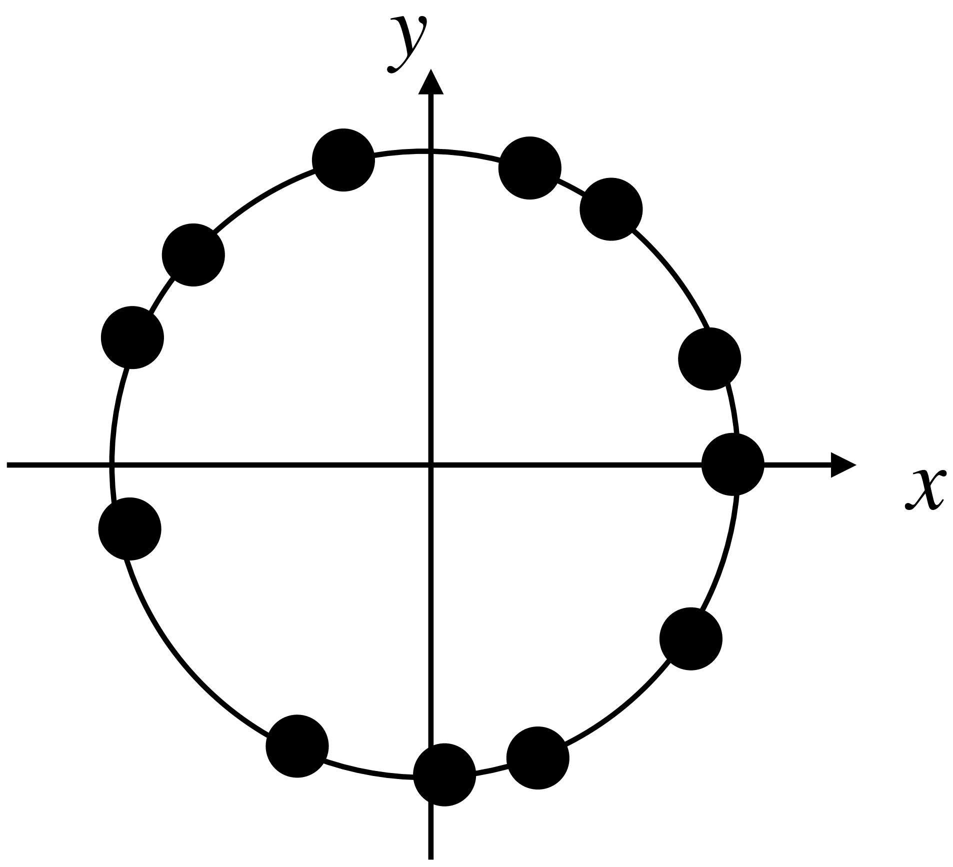
$(3-\sqrt{5})\pi$

Density implies rapid cover



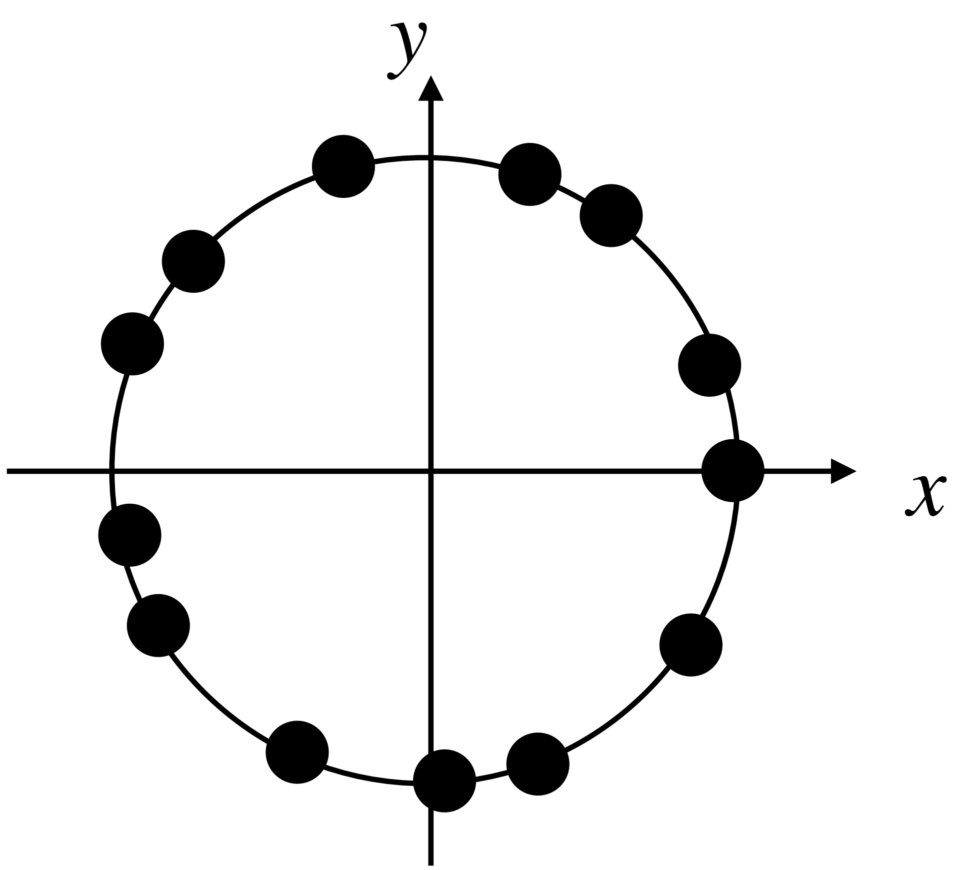
$$(3-\sqrt{5})\pi$$

Density implies rapid cover



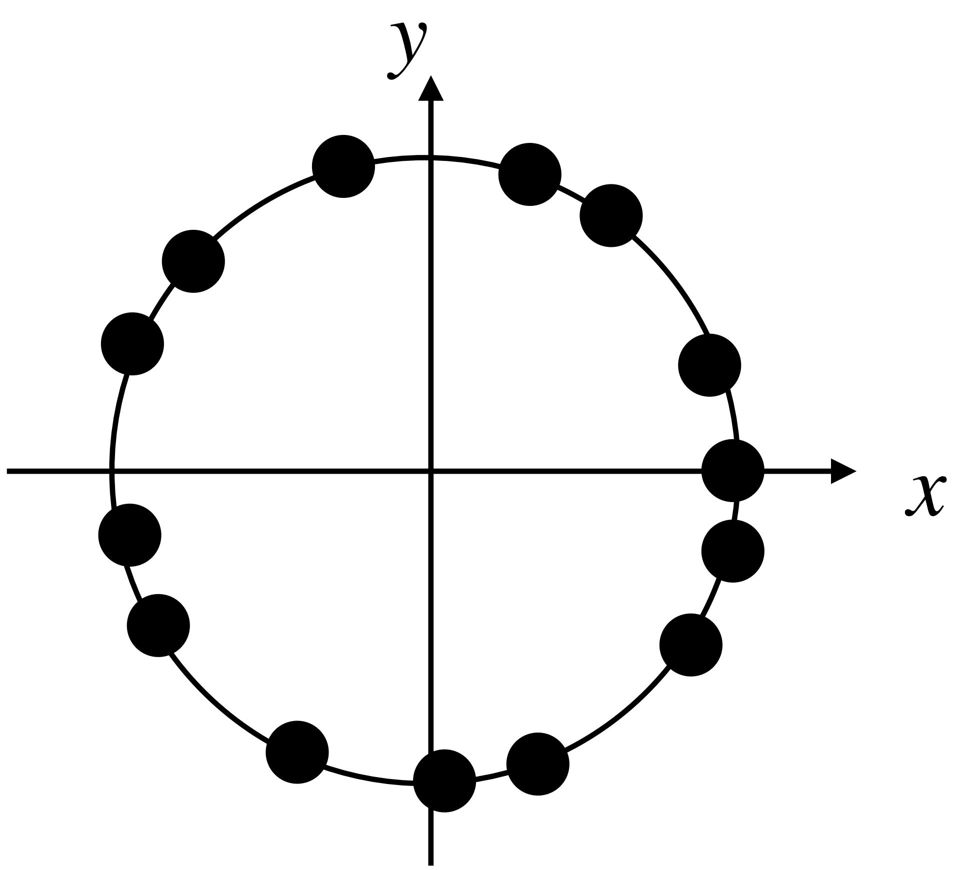
$$(3-\sqrt{5})\pi$$

Density implies rapid cover



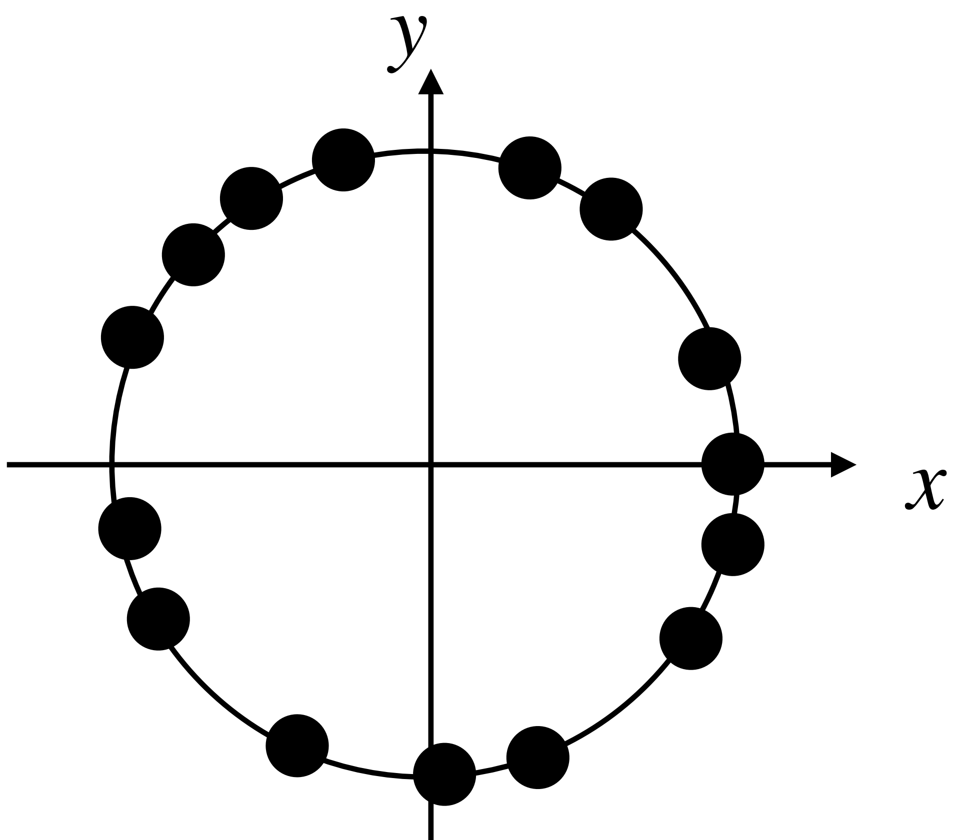
$(3 - \sqrt{5})\pi$

Density implies rapid cover



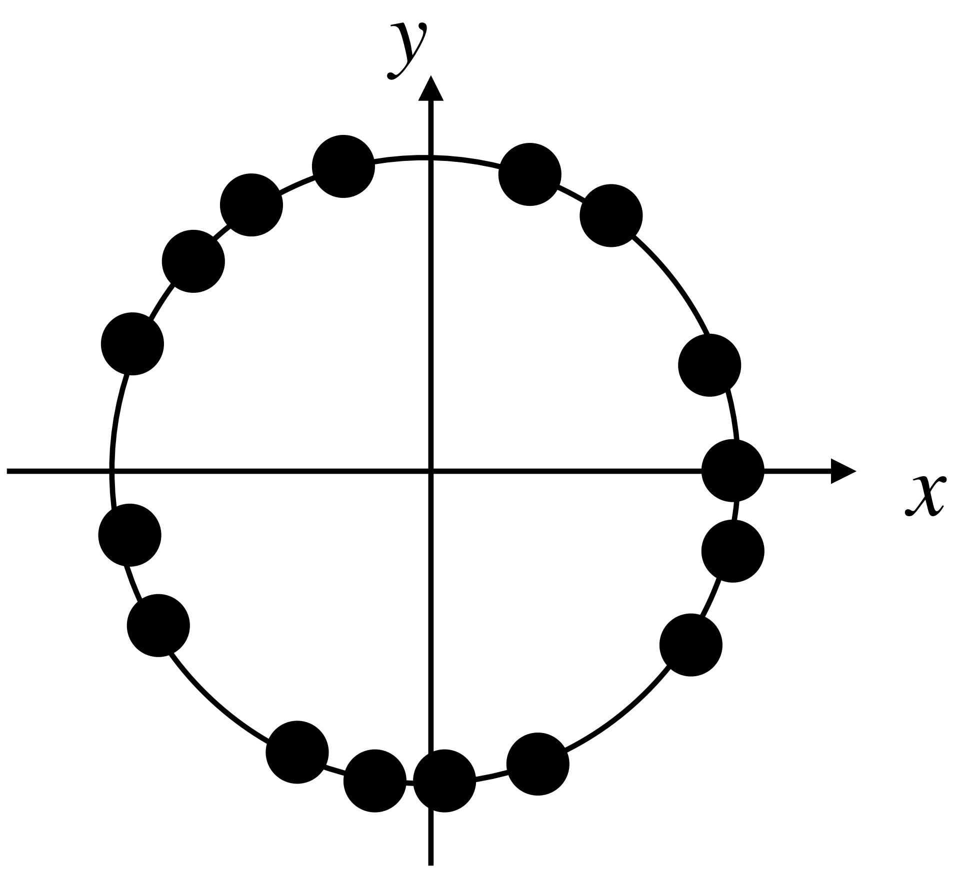
$$(3-\sqrt{5})\pi$$

Density implies rapid cover



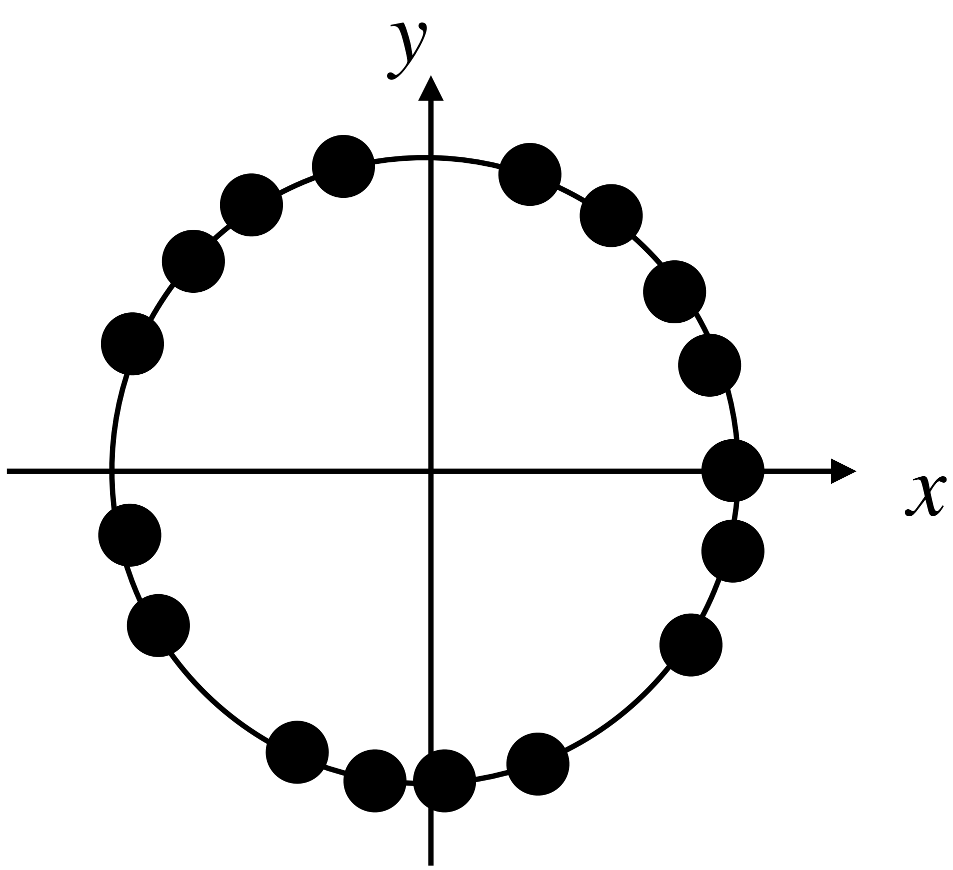
$(3-\sqrt{5})\pi$

Density implies rapid cover



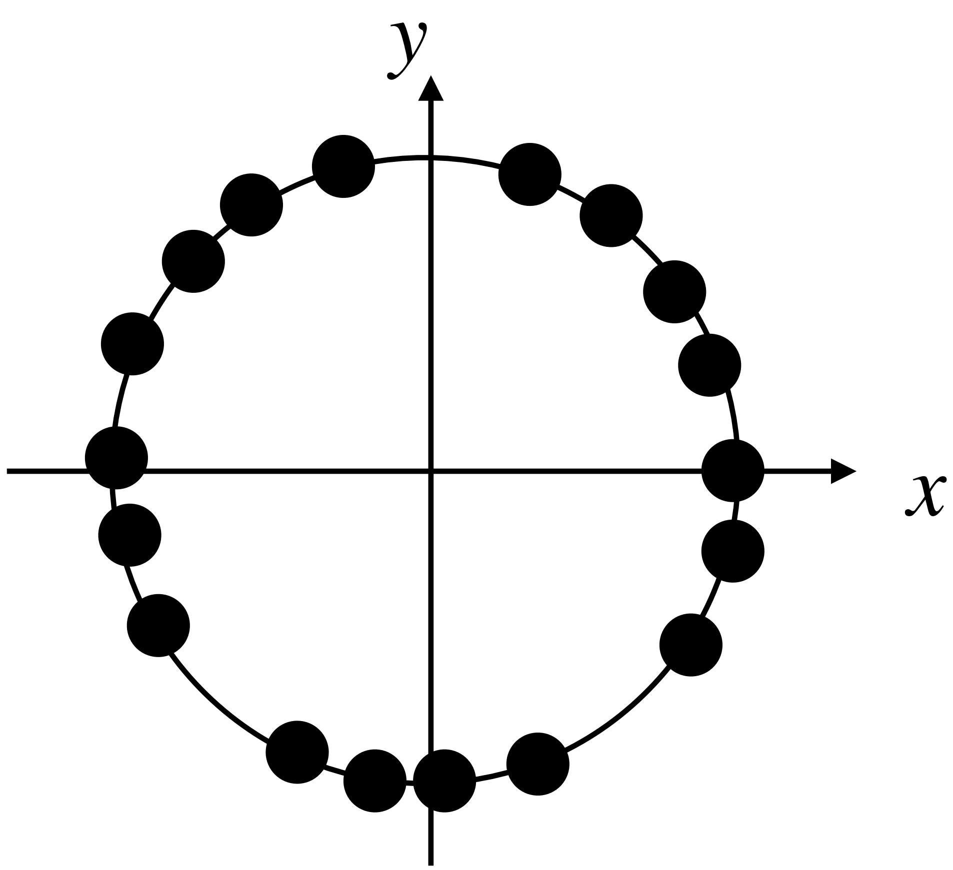
$(3-\sqrt{5})\pi$

Density implies rapid cover



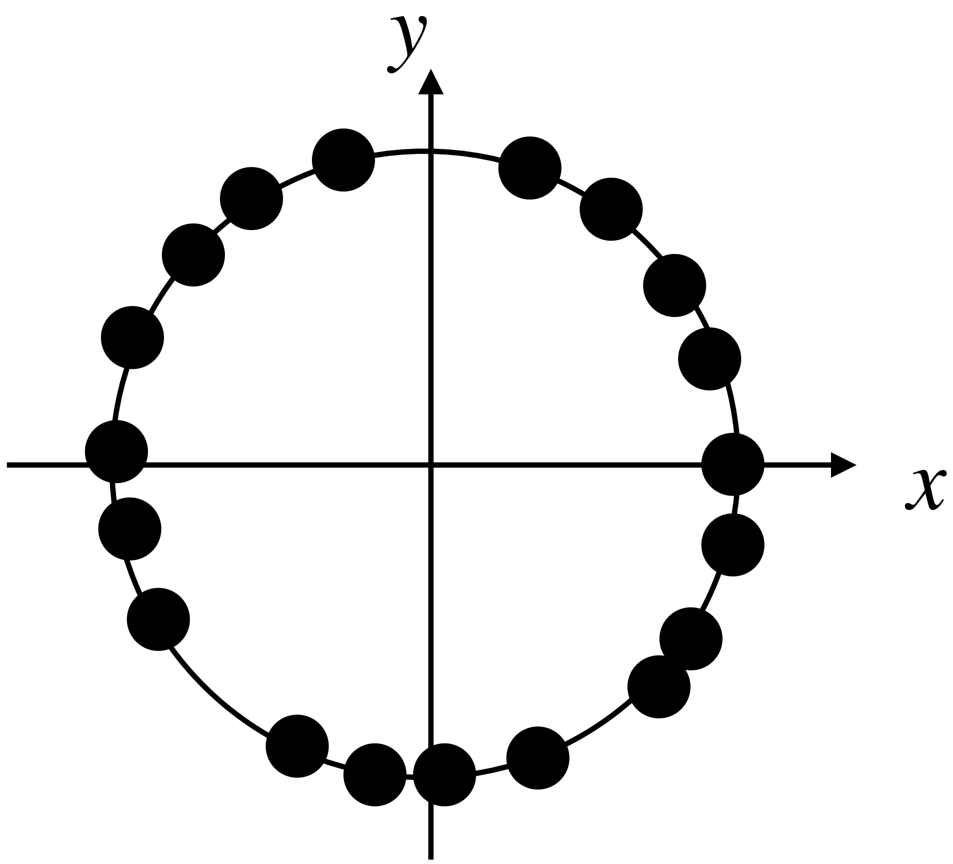
$$(3-\sqrt{5})\pi$$

Density implies rapid cover



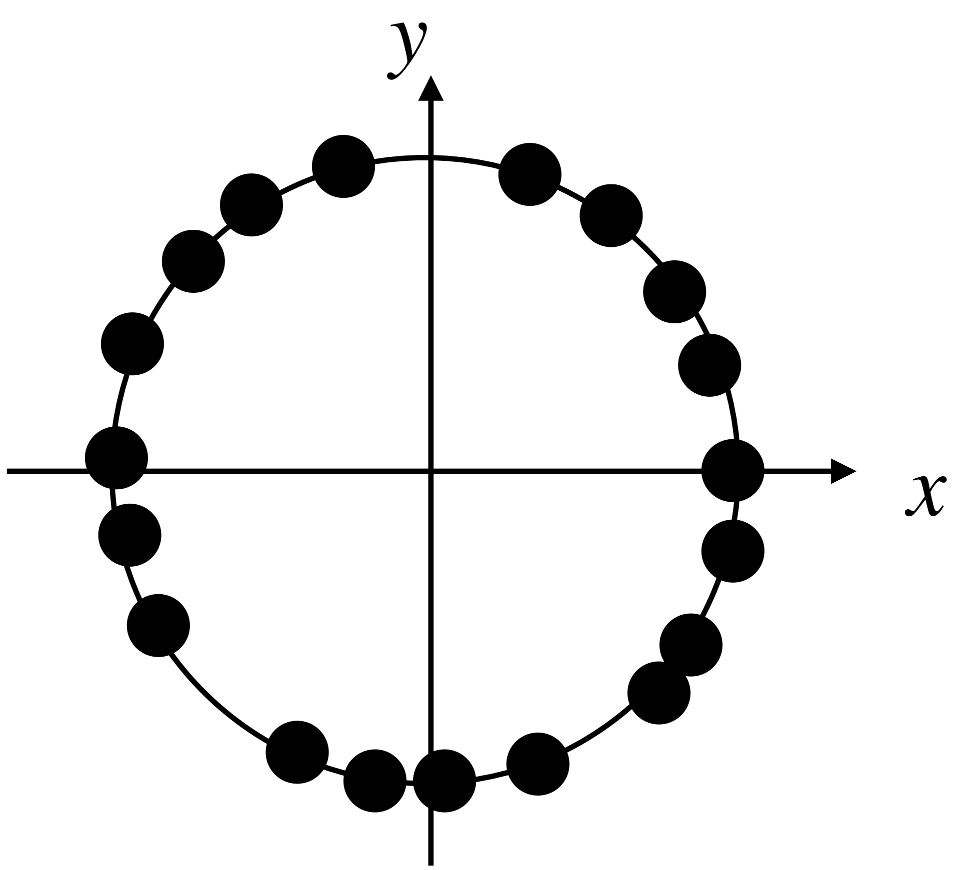
$(3-\sqrt{5})\pi$

Density implies rapid cover



$(3-\sqrt{5})\pi$

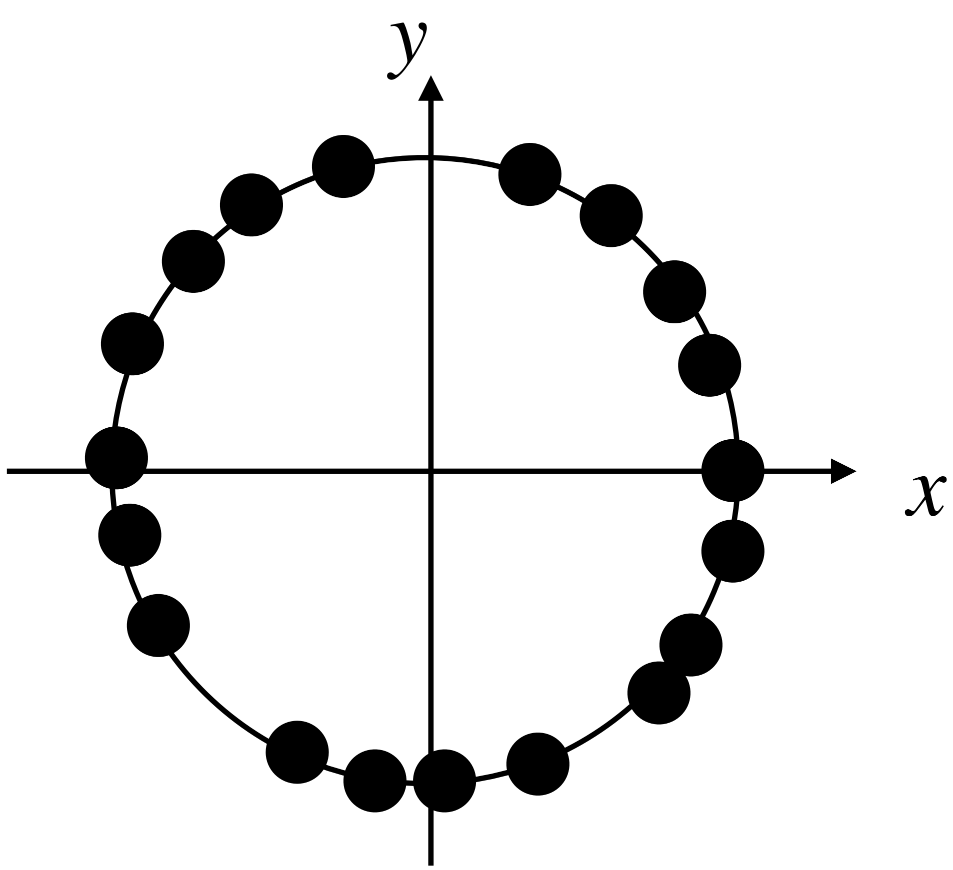
Density implies rapid cover



$$(3-\sqrt{5})\pi$$

An irrational angle to π can approximate an arbitrary angle rapidly.

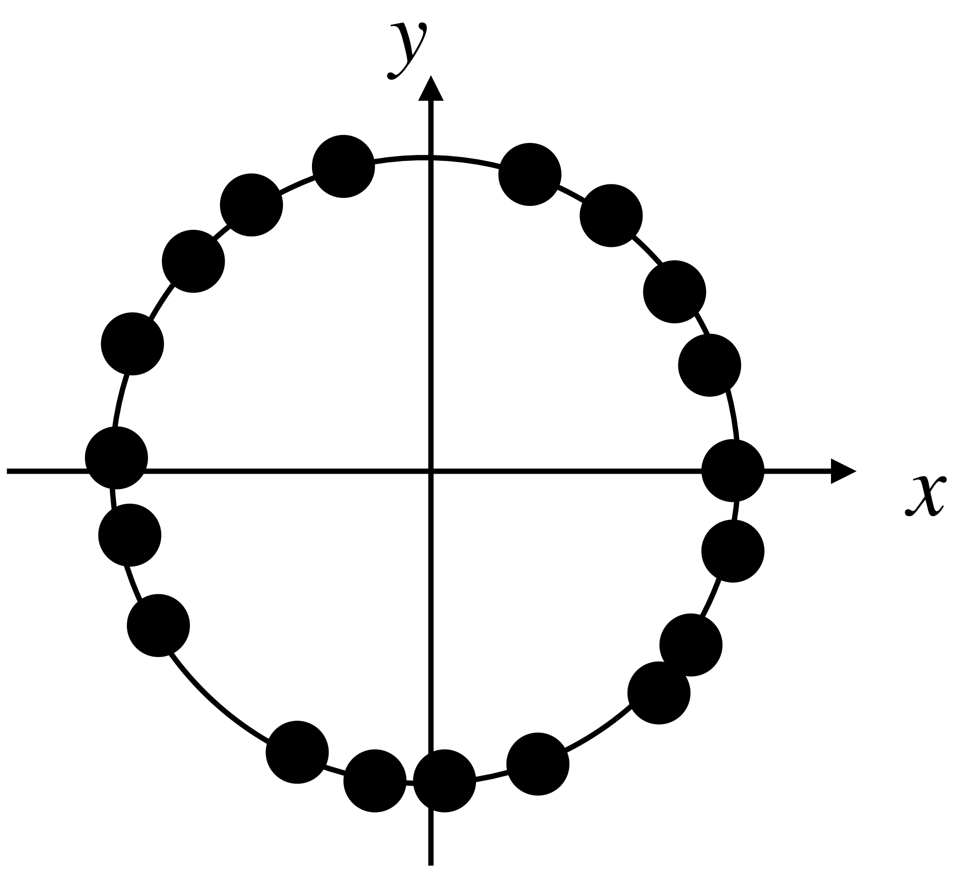
Density implies rapid cover



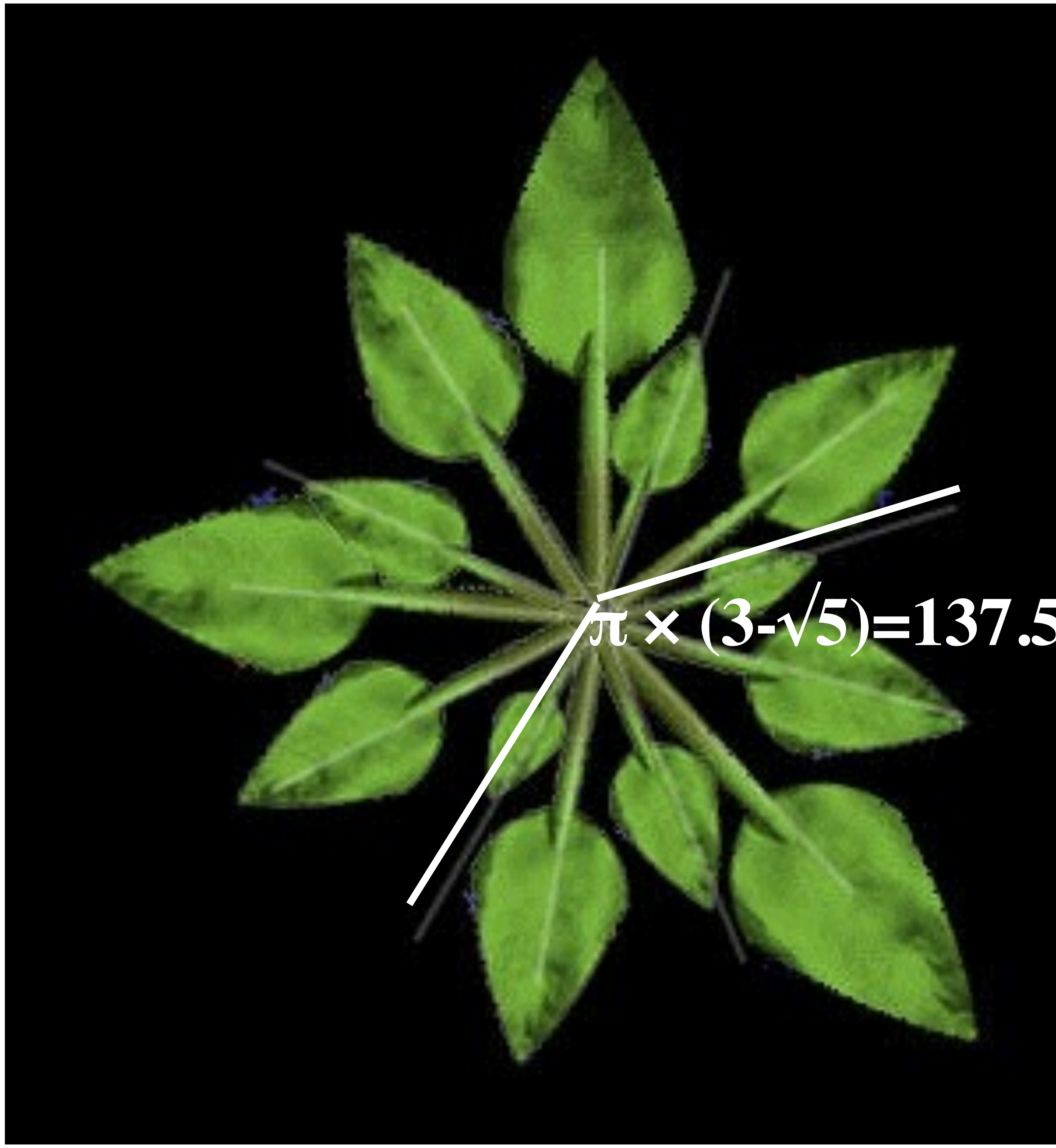
$$(3-\sqrt{5})\pi$$

An irrational angle to π can approximate an arbitrary angle rapidly.

Density implies rapid cover

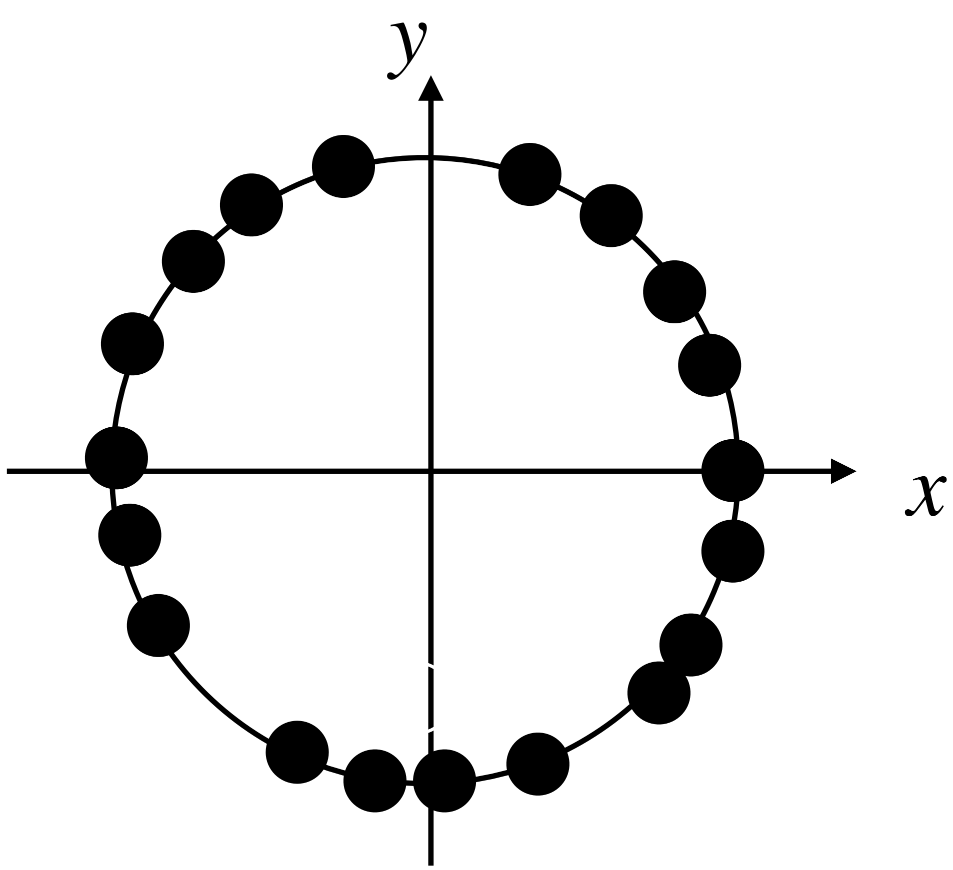


$$(3-\sqrt{5})\pi$$



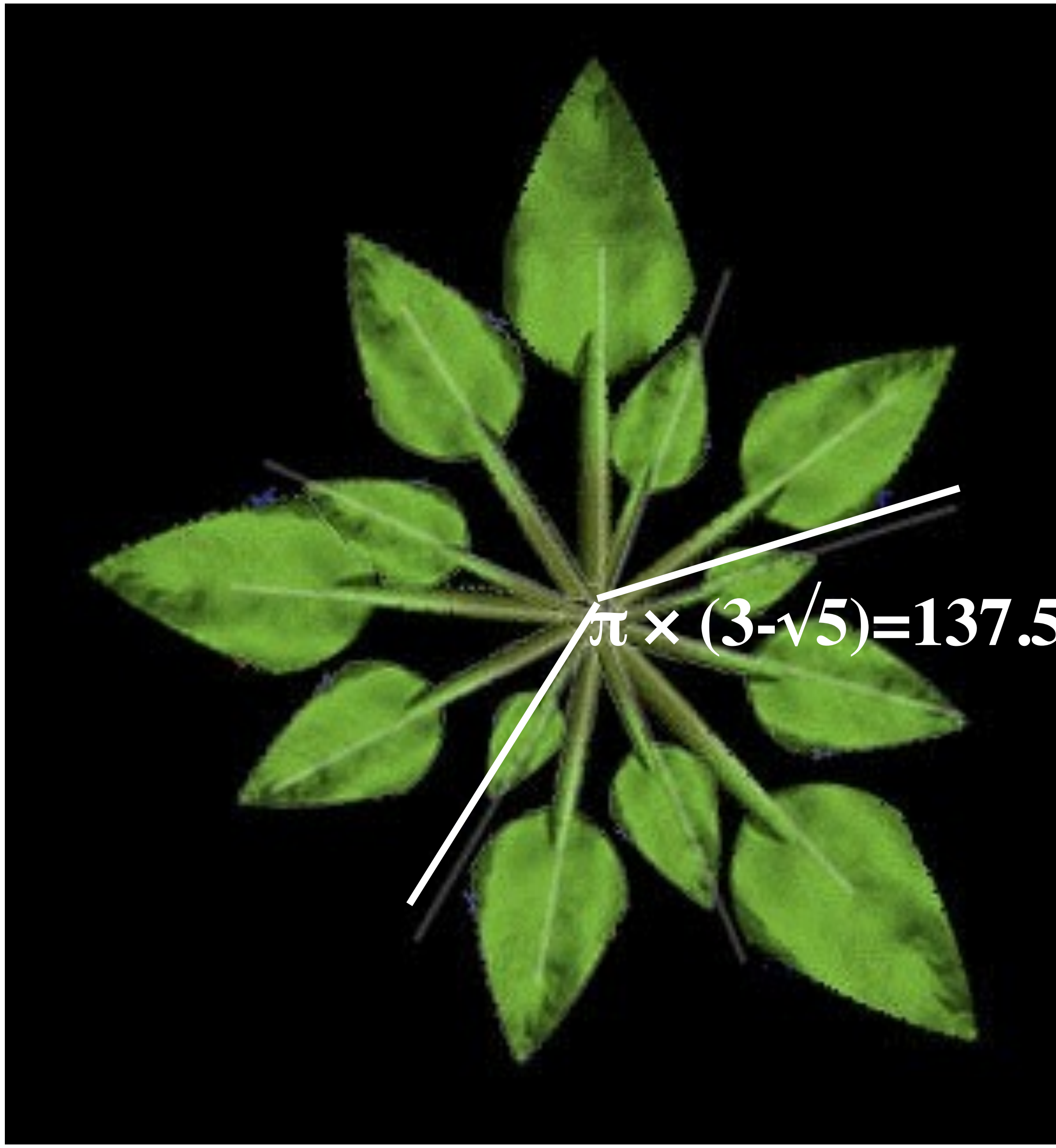
An irrational angle to π can approximate an arbitrary angle rapidly.

Density implies rapid cover

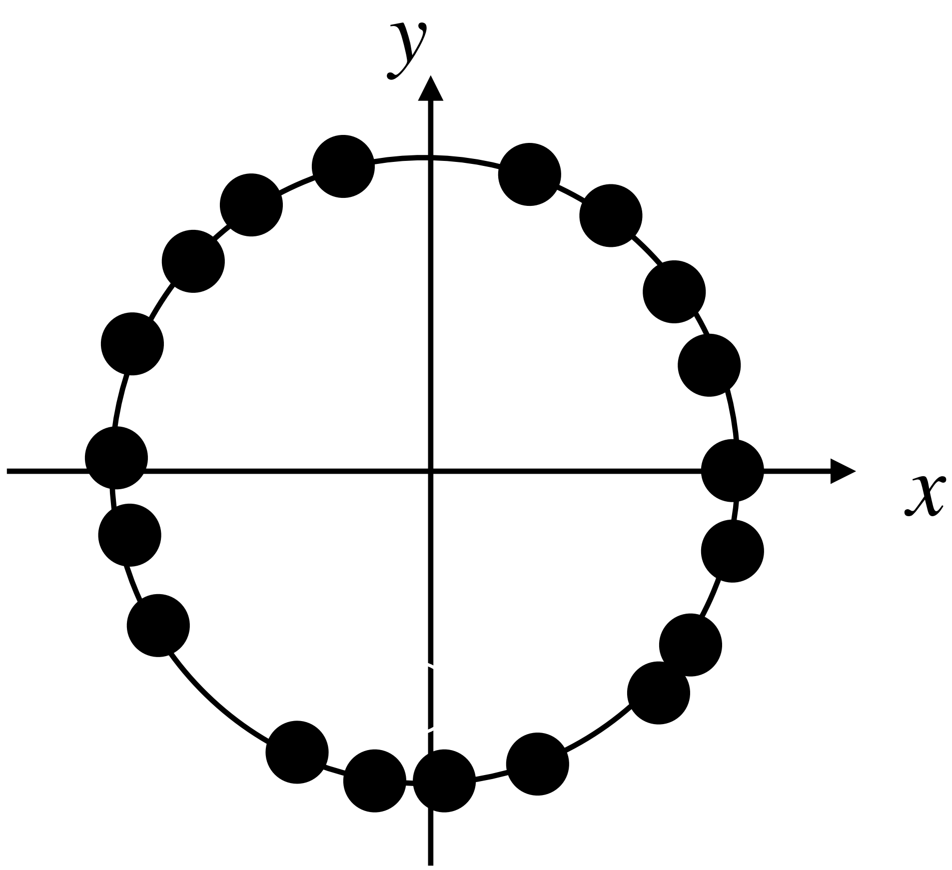


$$(3-\sqrt{5})\pi$$

An irrational angle to π can approximate an arbitrary angle rapidly.



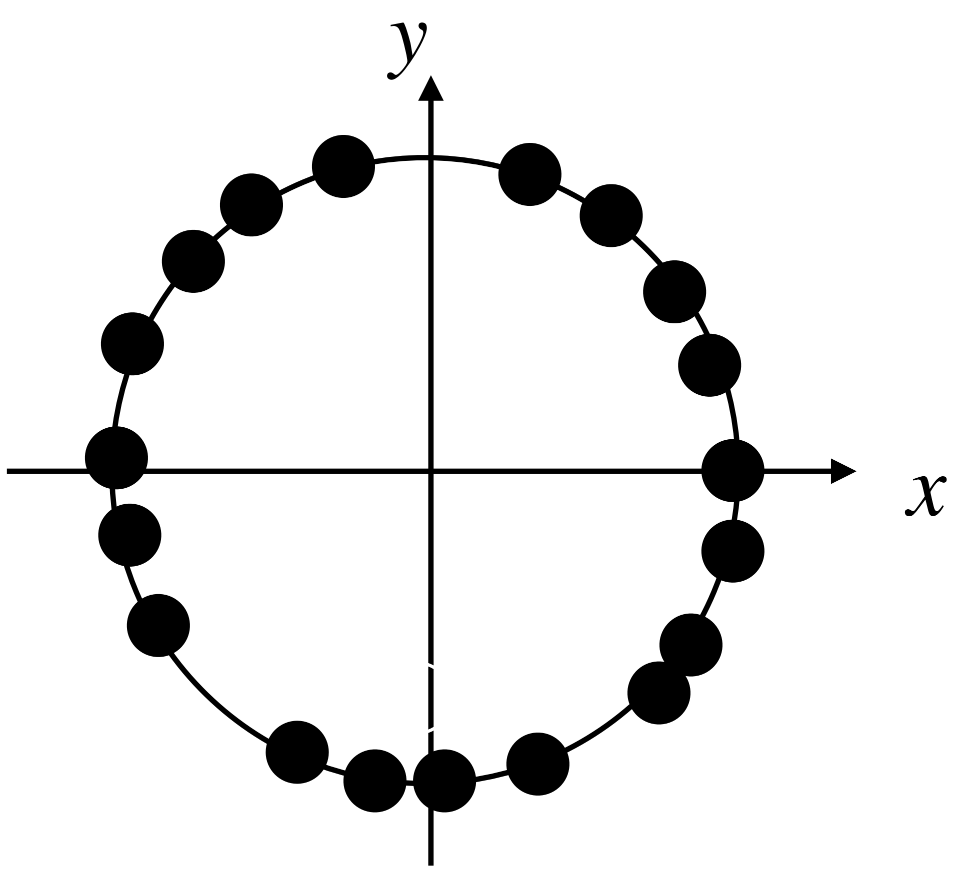
Density implies rapid cover



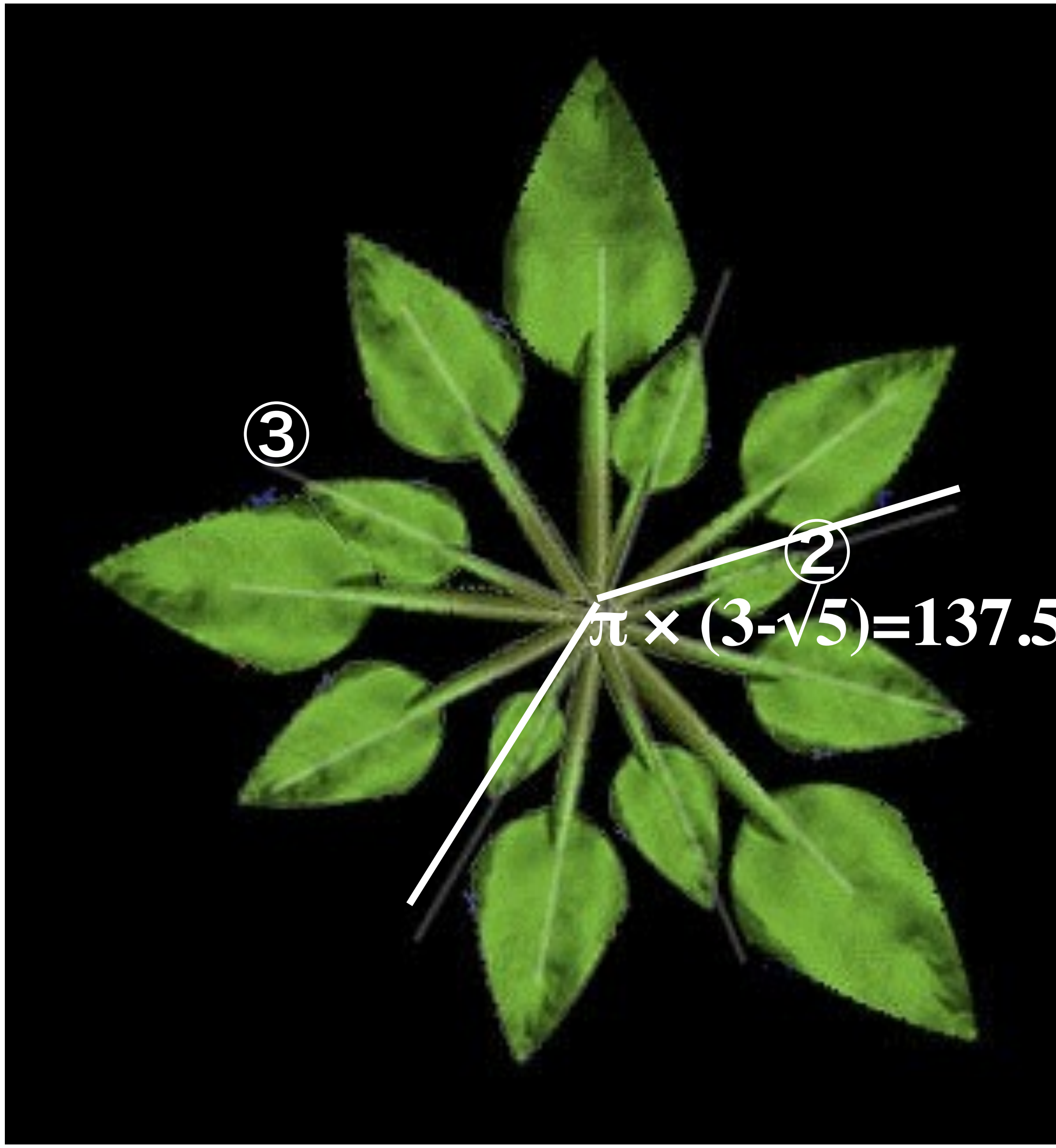
$$(3-\sqrt{5})\pi$$

An irrational angle to π can approximate an arbitrary angle rapidly.

Density implies rapid cover

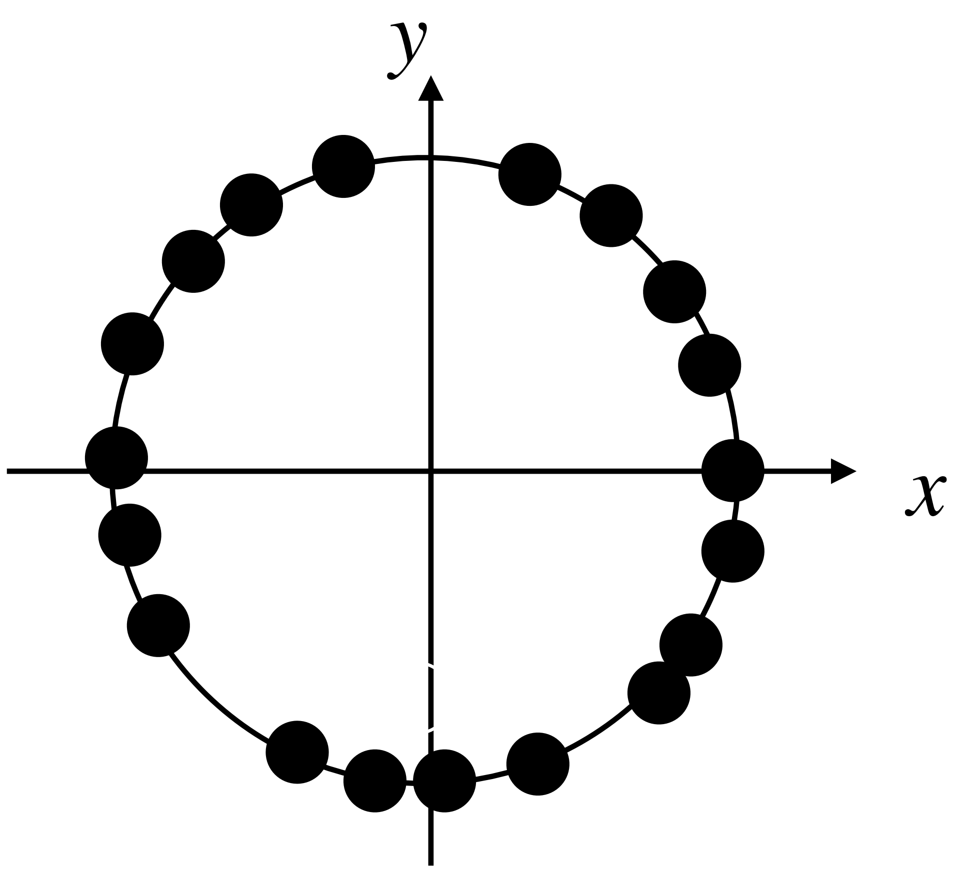


$$(3-\sqrt{5})\pi$$

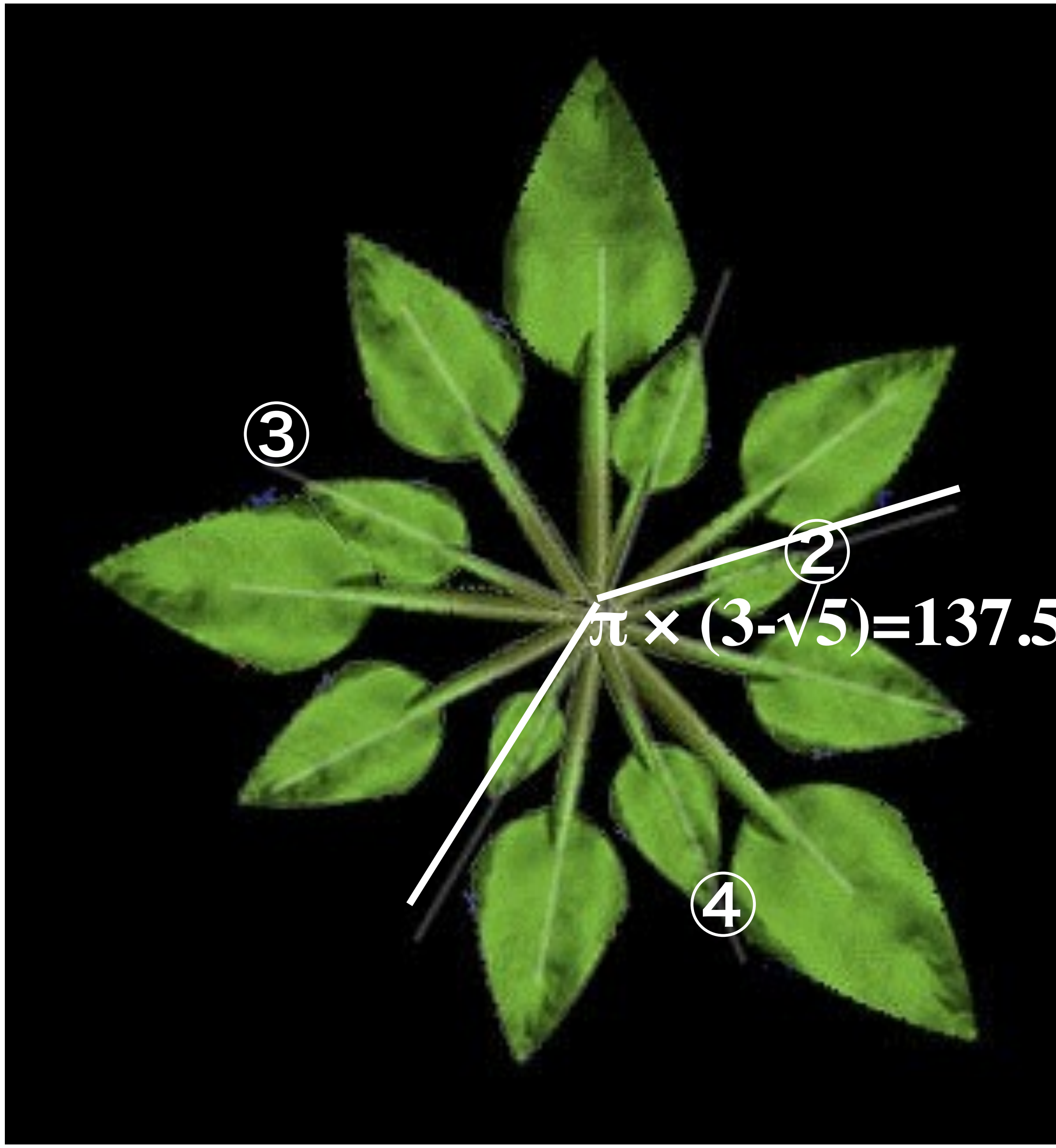


An irrational angle to π can approximate an arbitrary angle rapidly.

Density implies rapid cover

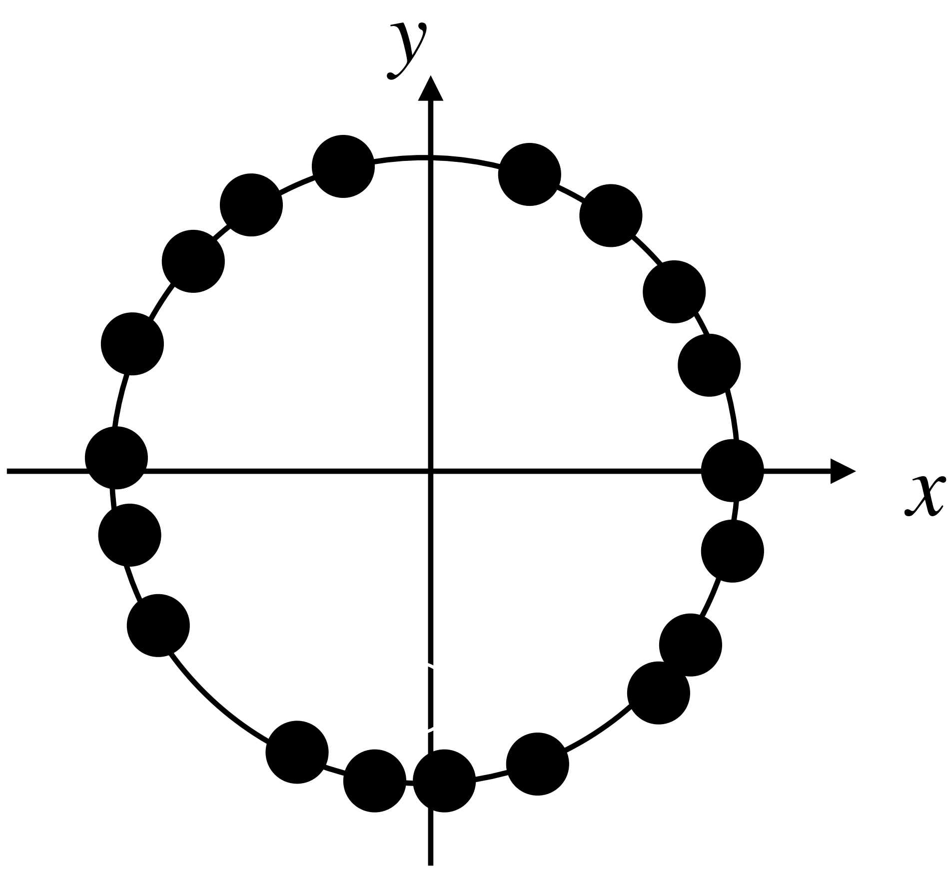


$$(3-\sqrt{5})\pi$$



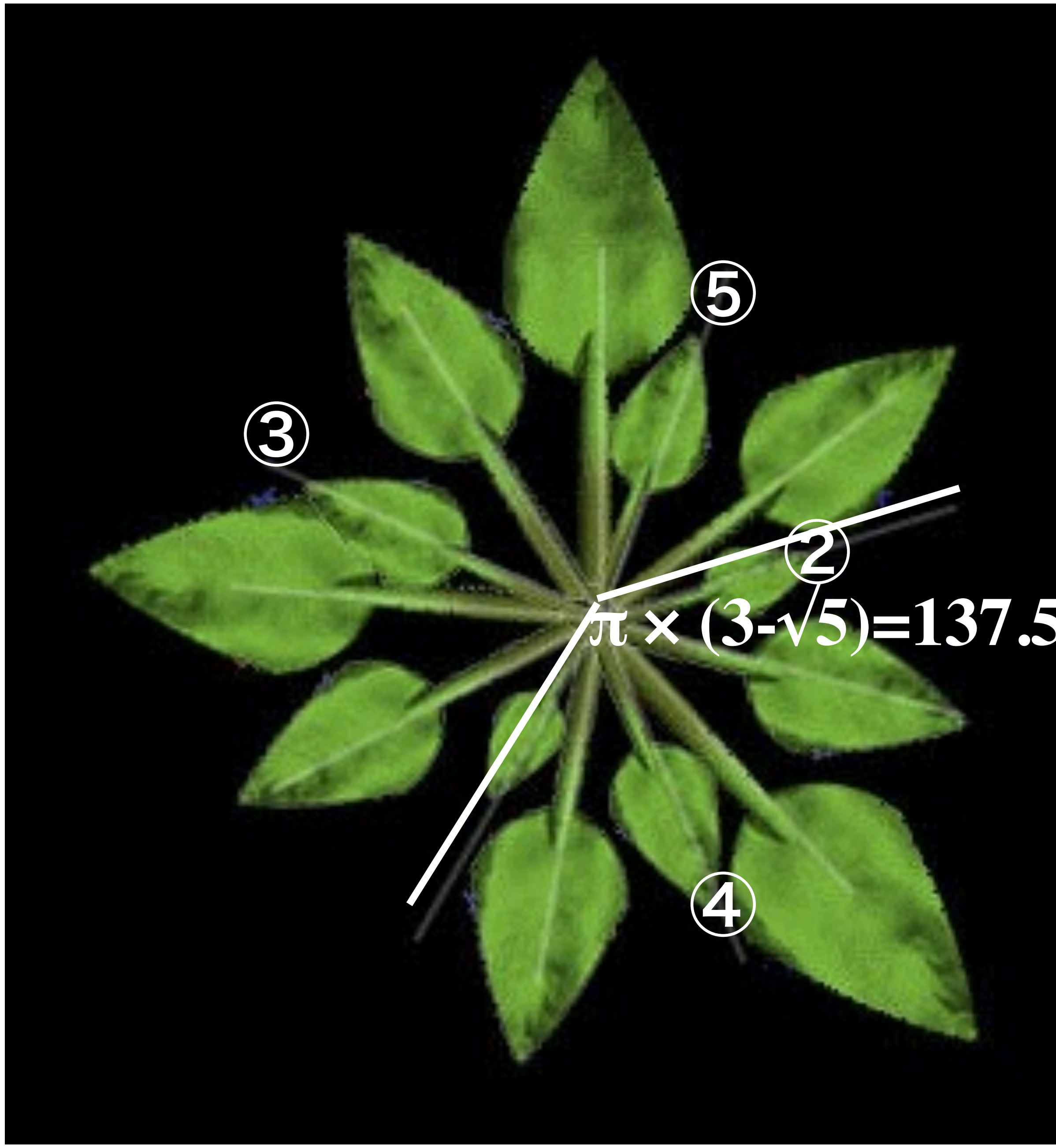
An irrational angle to π can approximate an arbitrary angle rapidly.

Density implies rapid cover

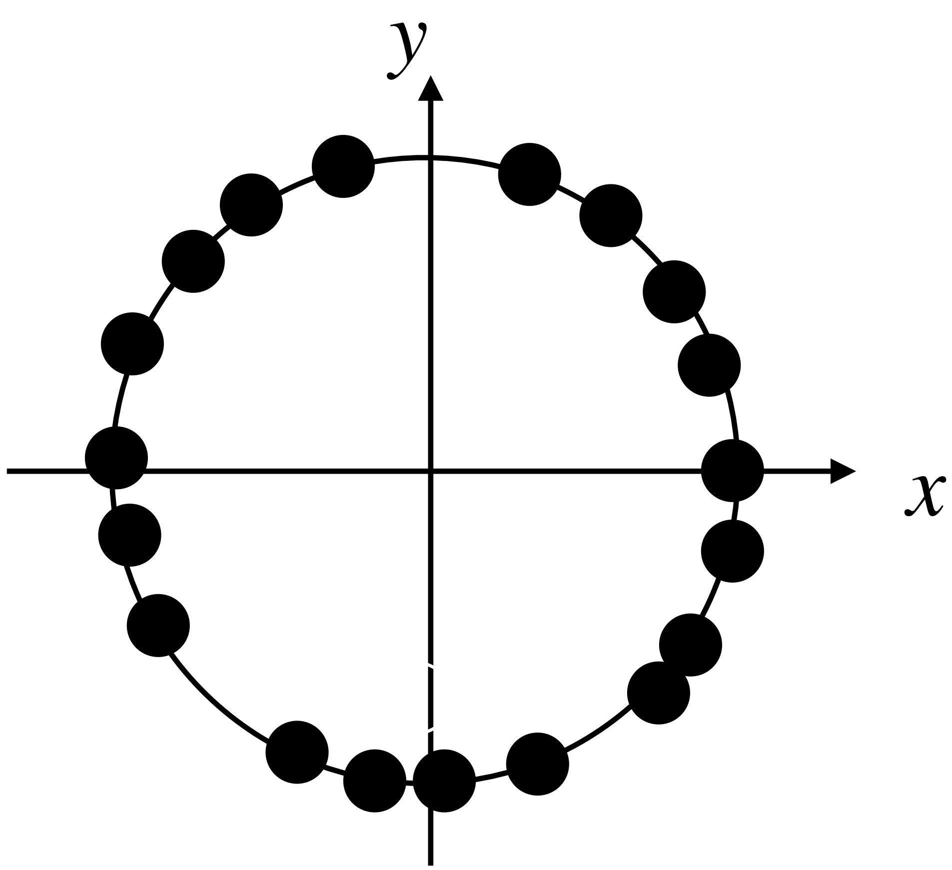


$$(3-\sqrt{5})\pi$$

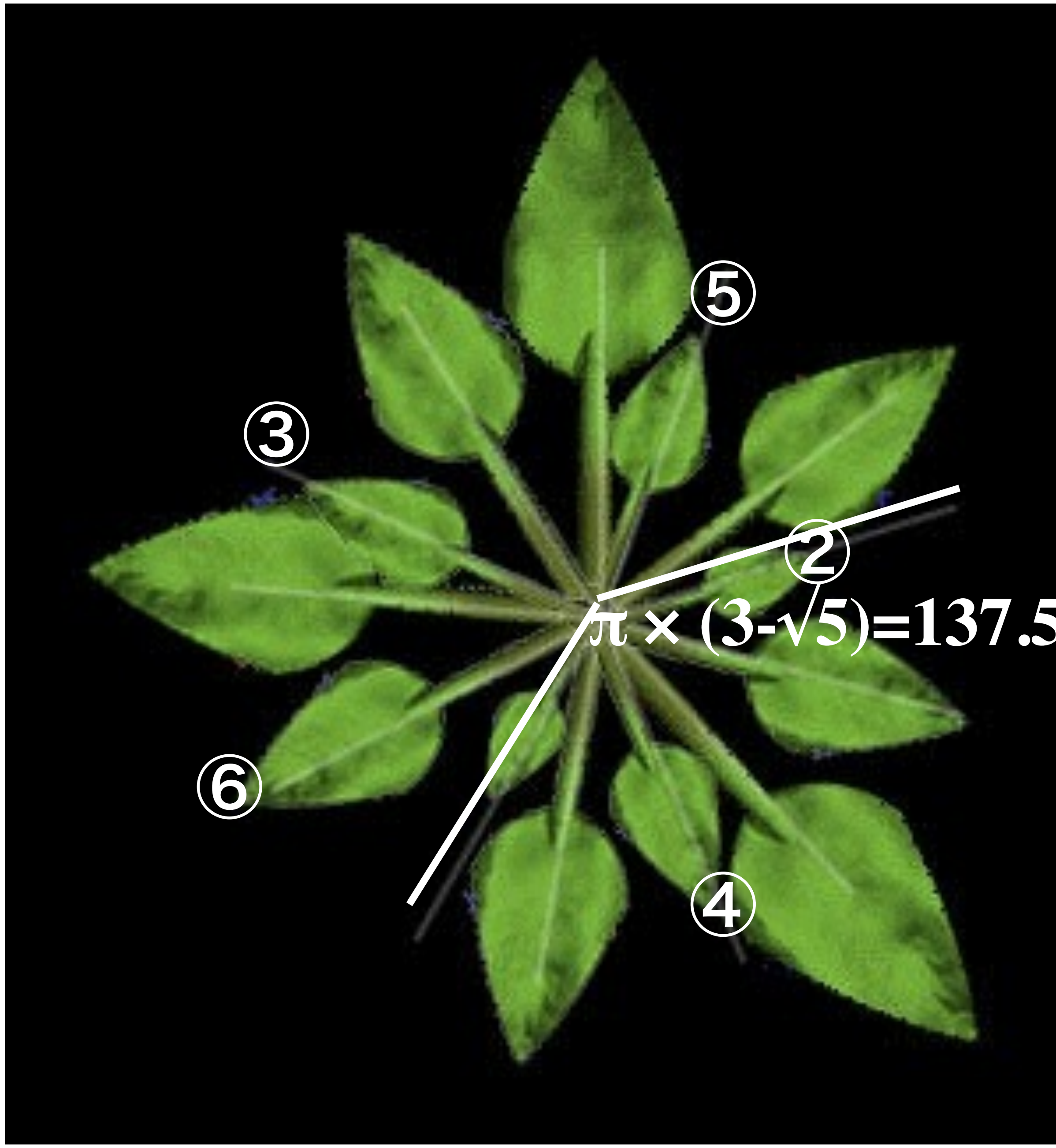
An irrational angle to π can approximate an arbitrary angle rapidly.



Density implies rapid cover

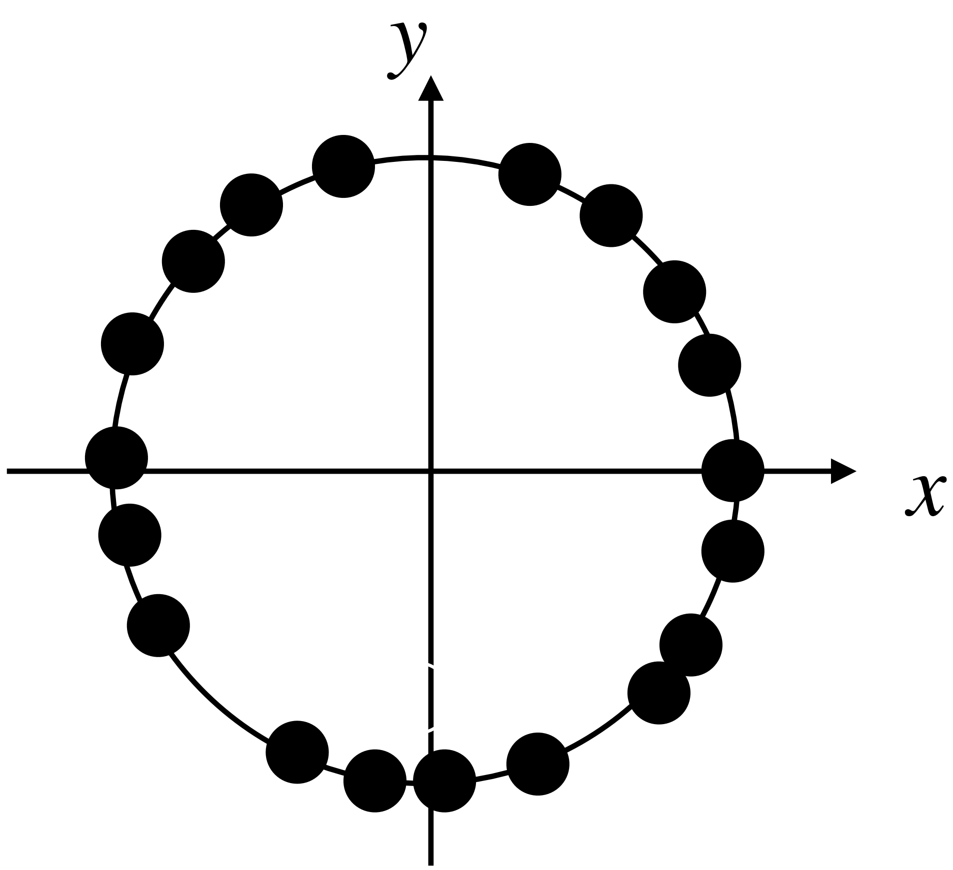


$$(3-\sqrt{5})\pi$$



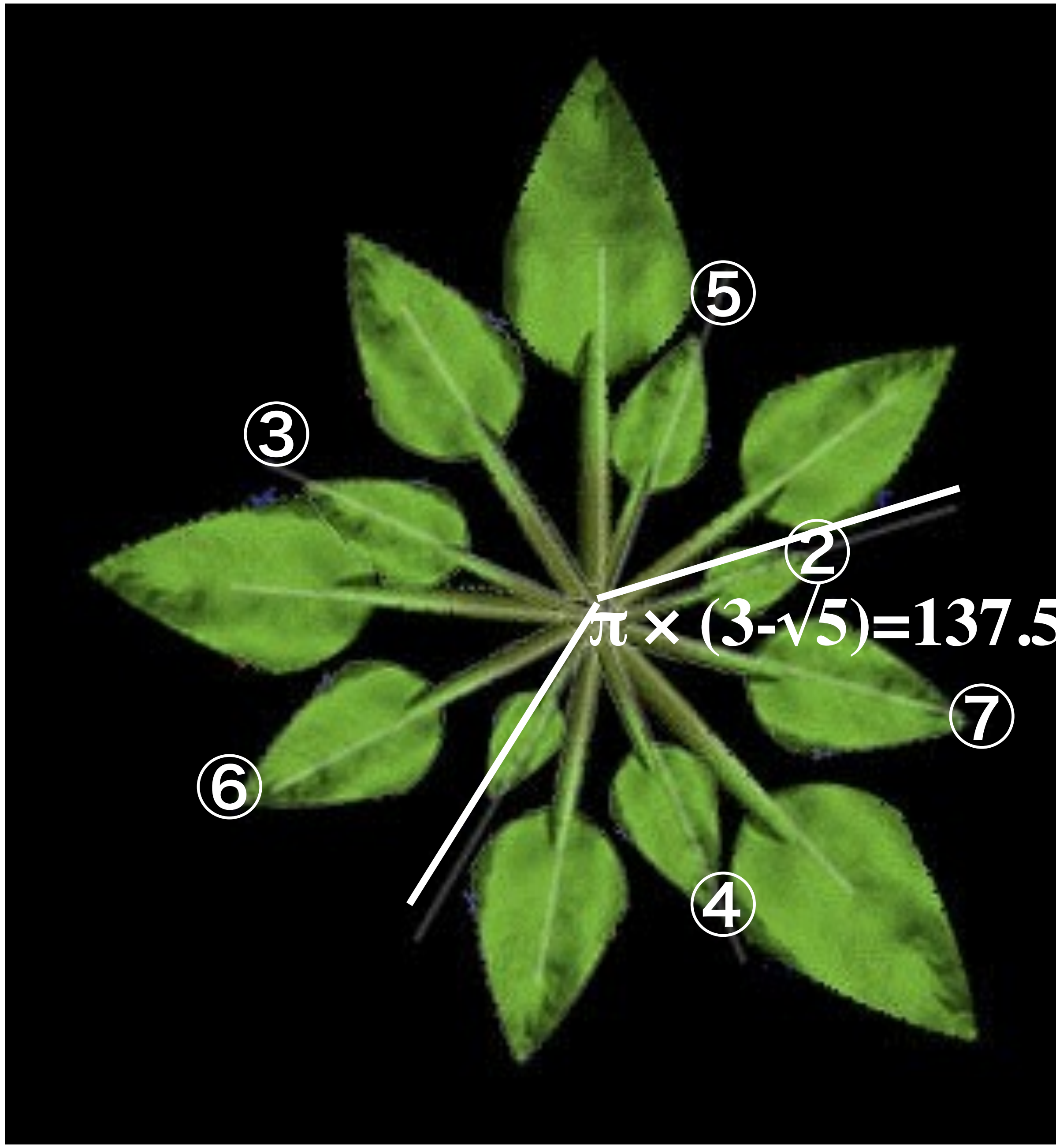
An irrational angle to π can approximate an arbitrary angle rapidly.

Density implies rapid cover

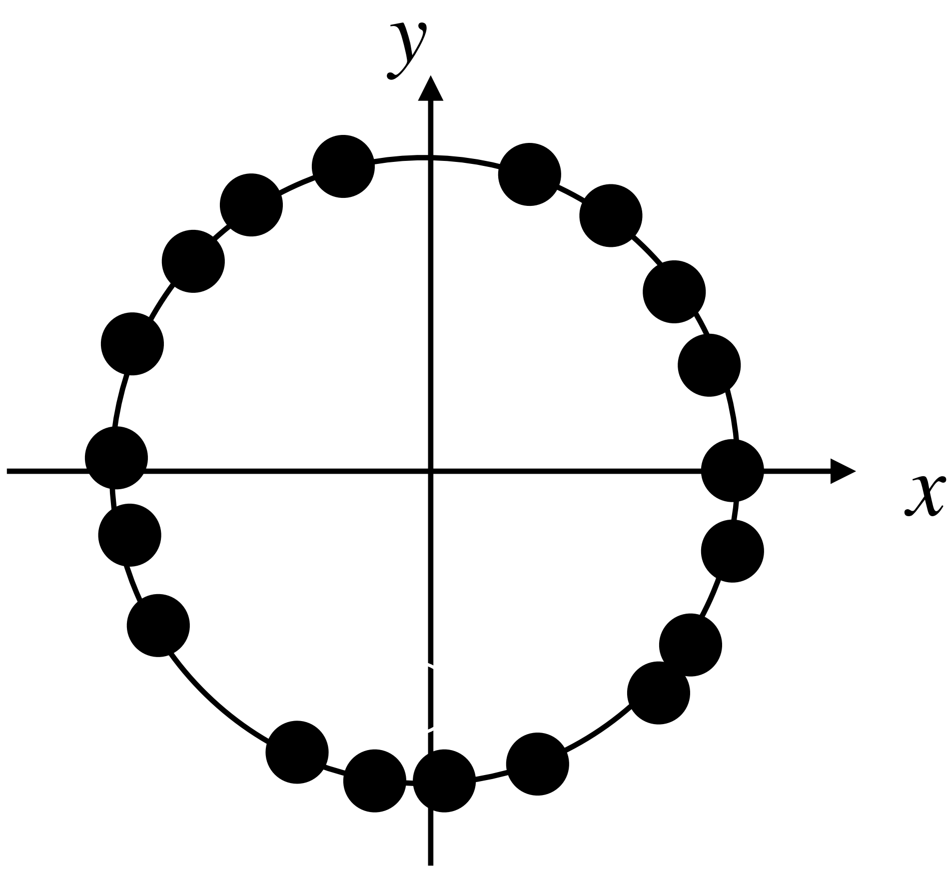


$$(3-\sqrt{5})\pi$$

An irrational angle to π can approximate an arbitrary angle rapidly.

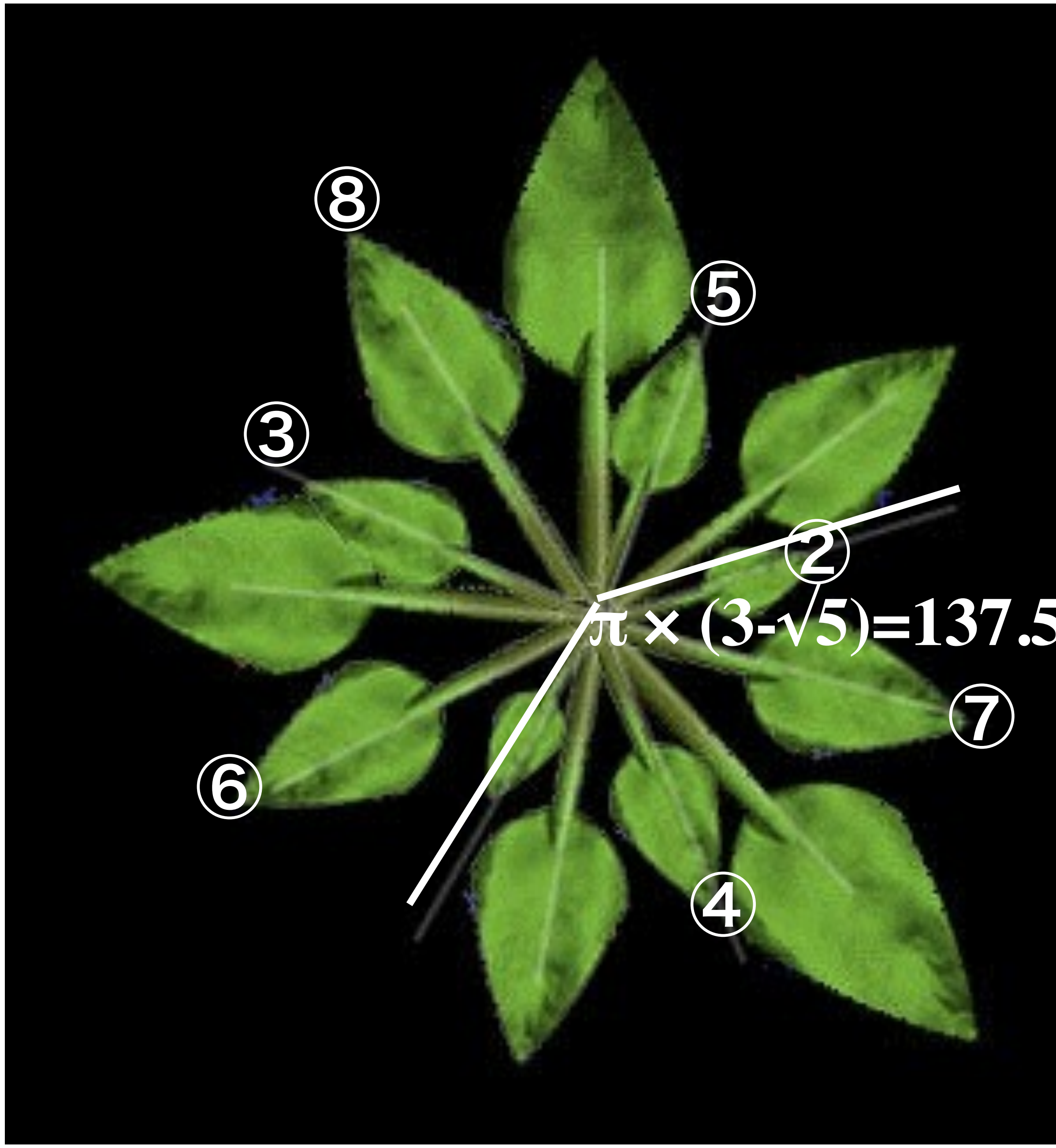


Density implies rapid cover

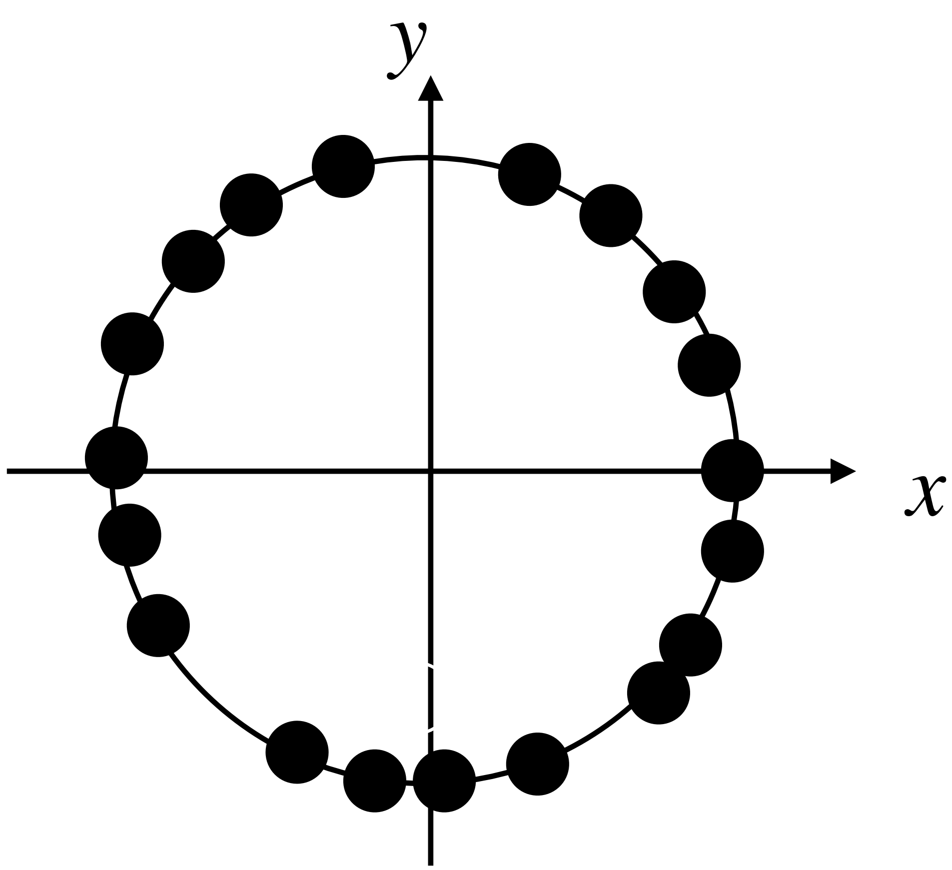


$$(3-\sqrt{5})\pi$$

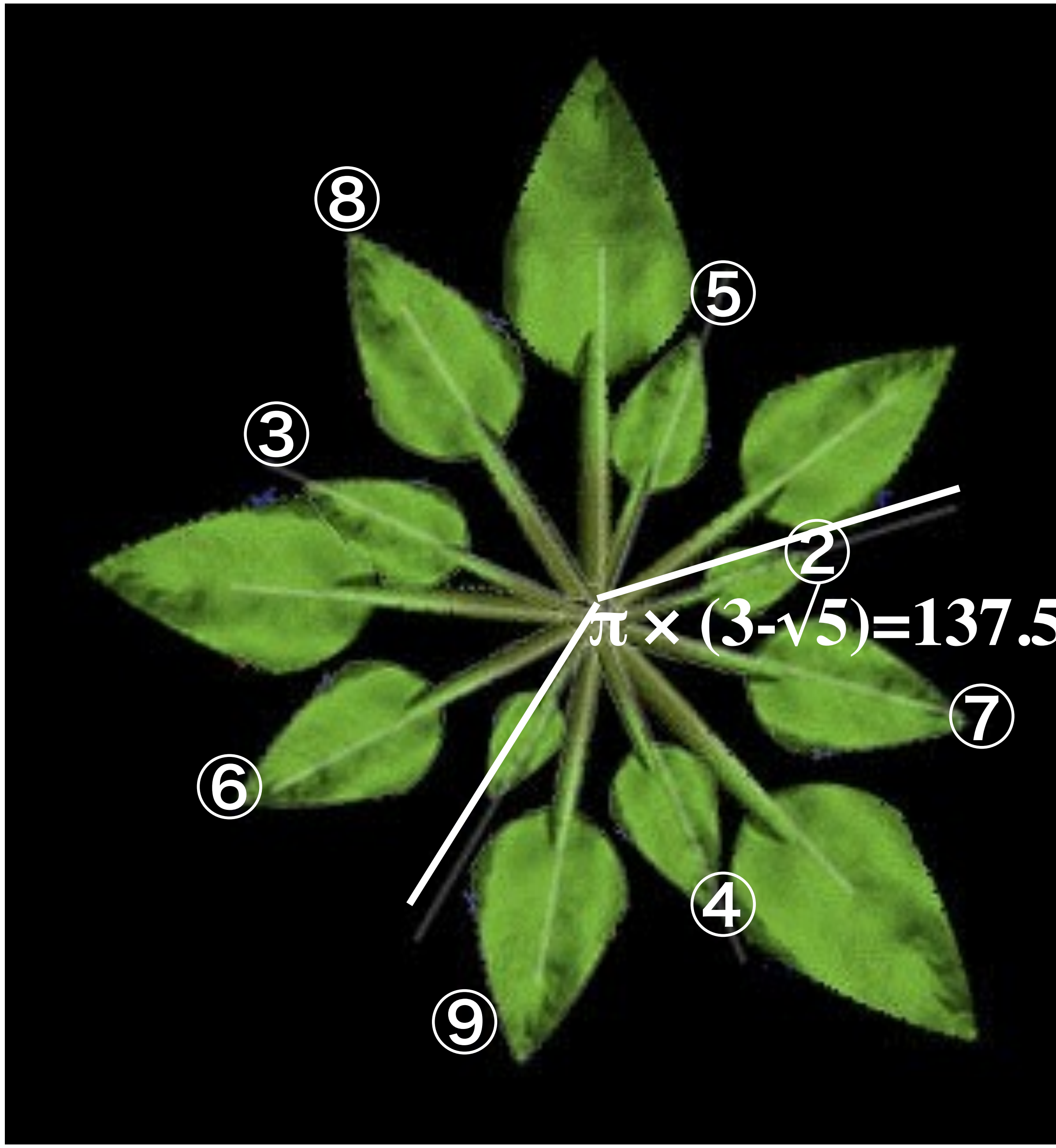
An irrational angle to π can approximate an arbitrary angle rapidly.



Density implies rapid cover

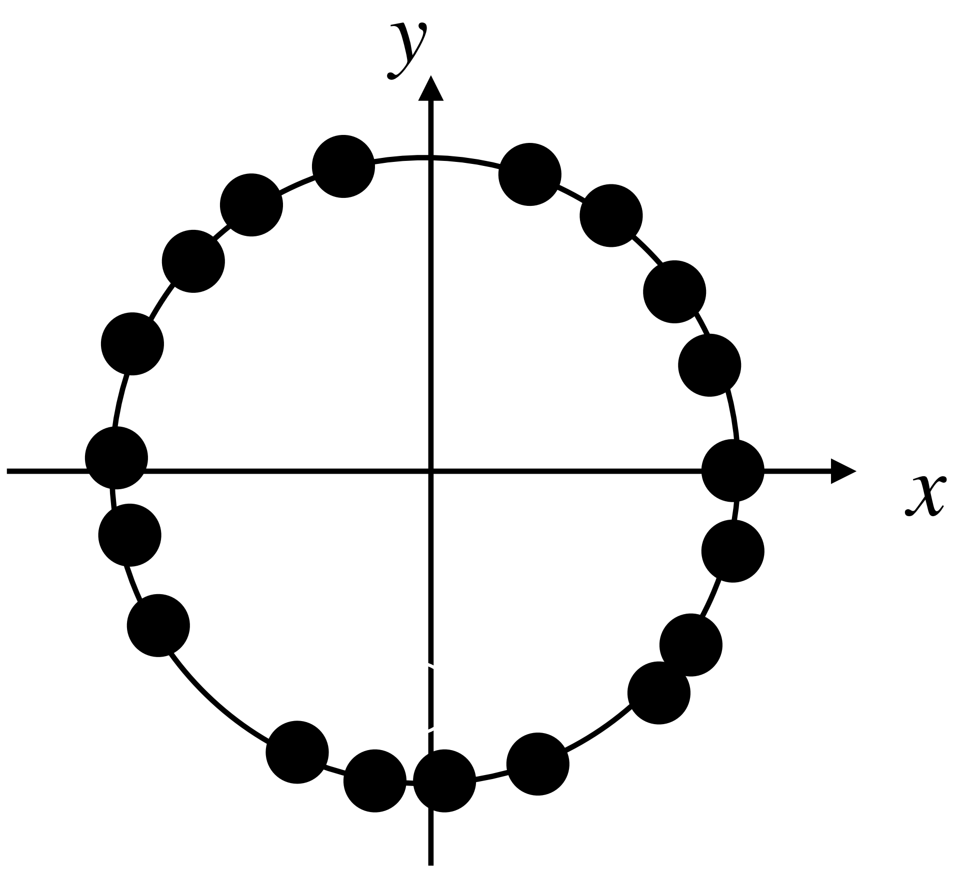


$$(3-\sqrt{5})\pi$$

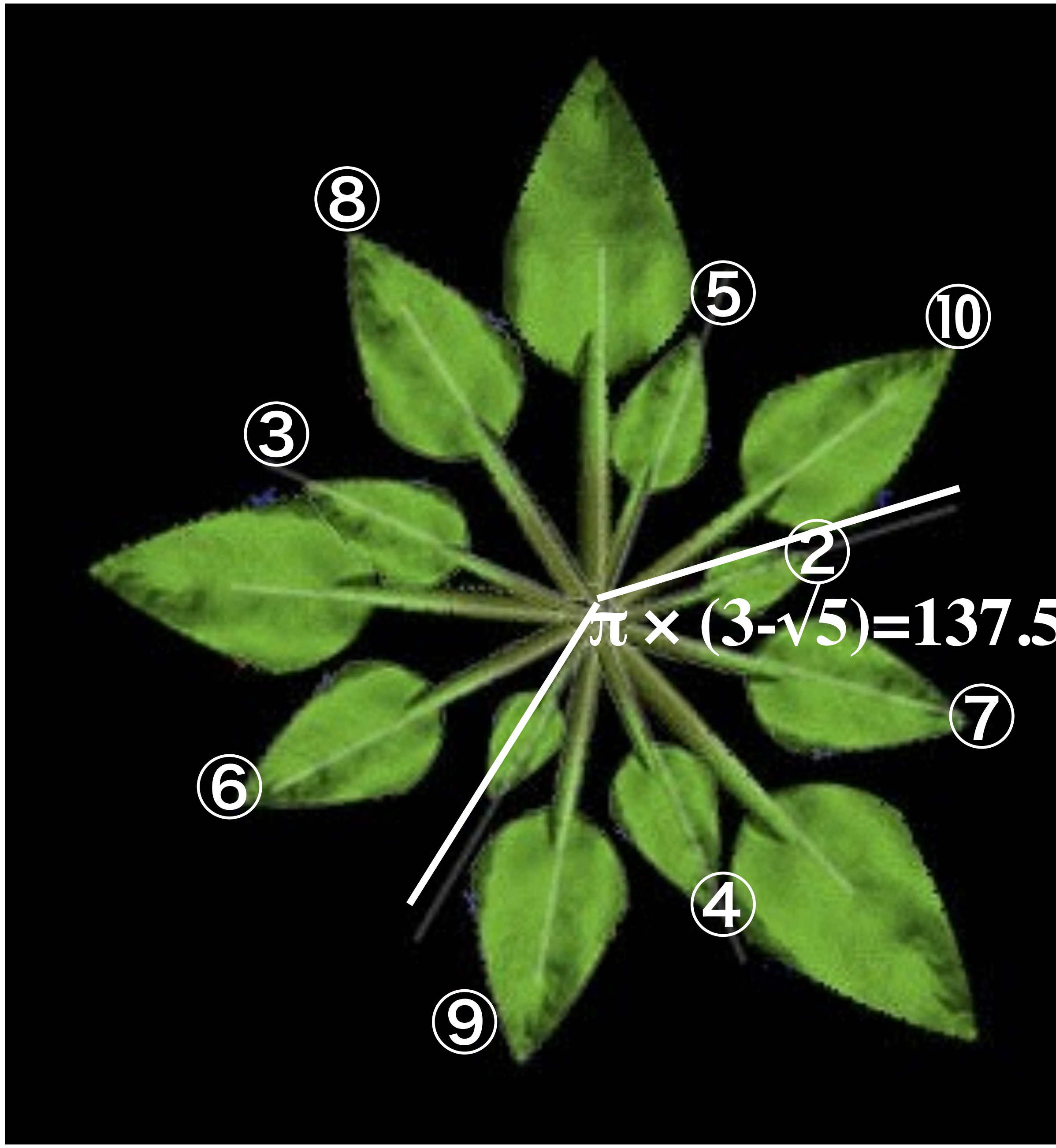


An irrational angle to π can approximate an arbitrary angle rapidly.

Density implies rapid cover

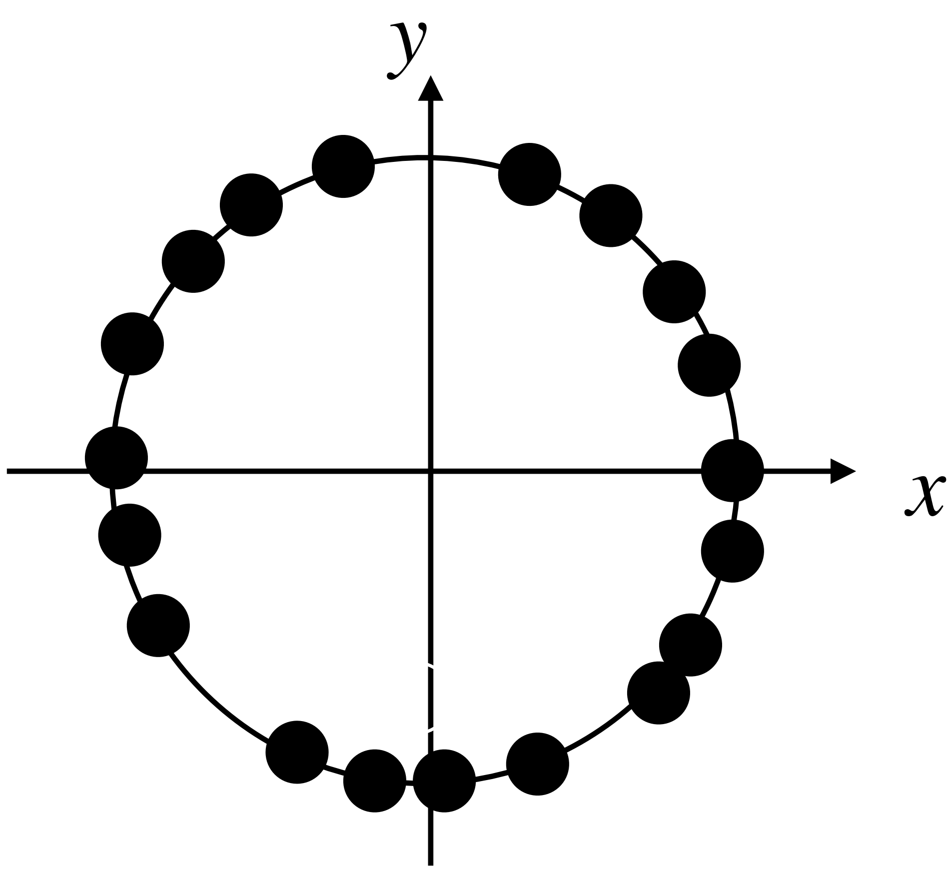


$$(3-\sqrt{5})\pi$$



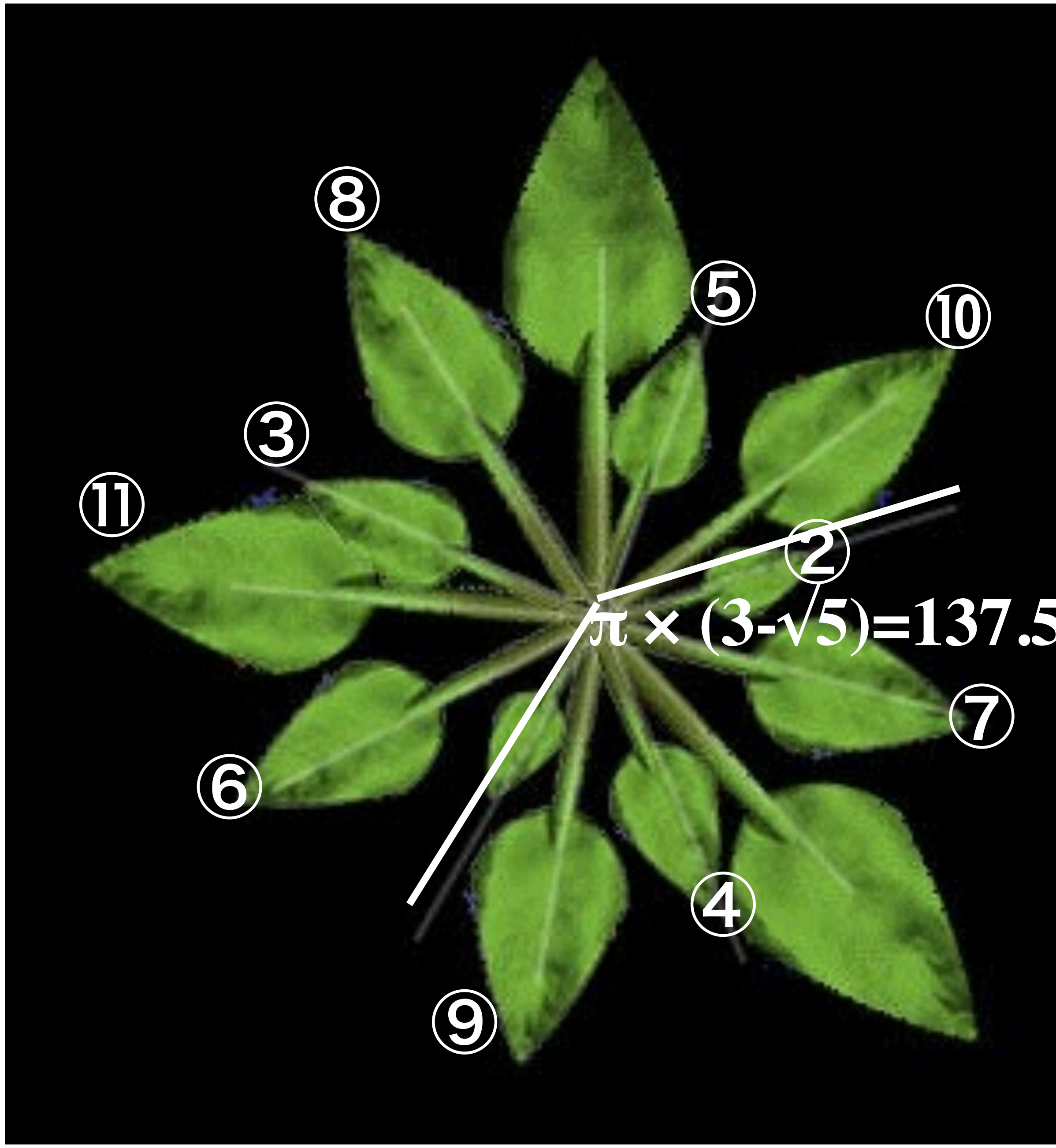
An irrational angle to π can approximate an arbitrary angle rapidly.

Density implies rapid cover

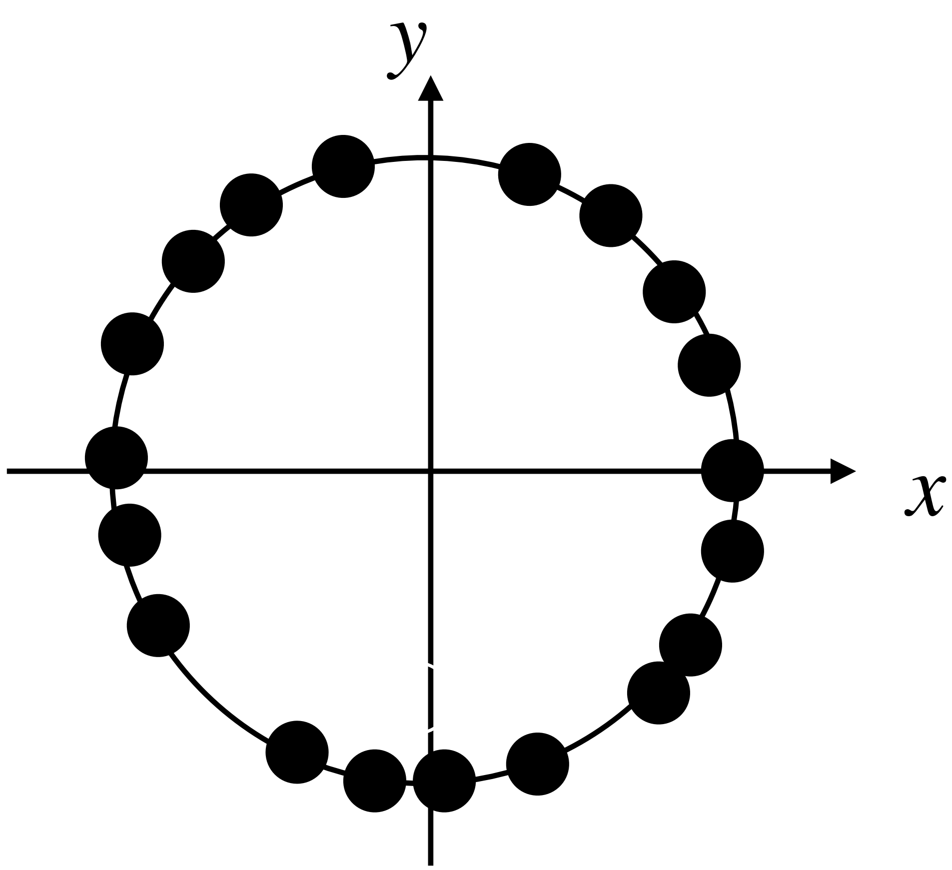


$$(3-\sqrt{5})\pi$$

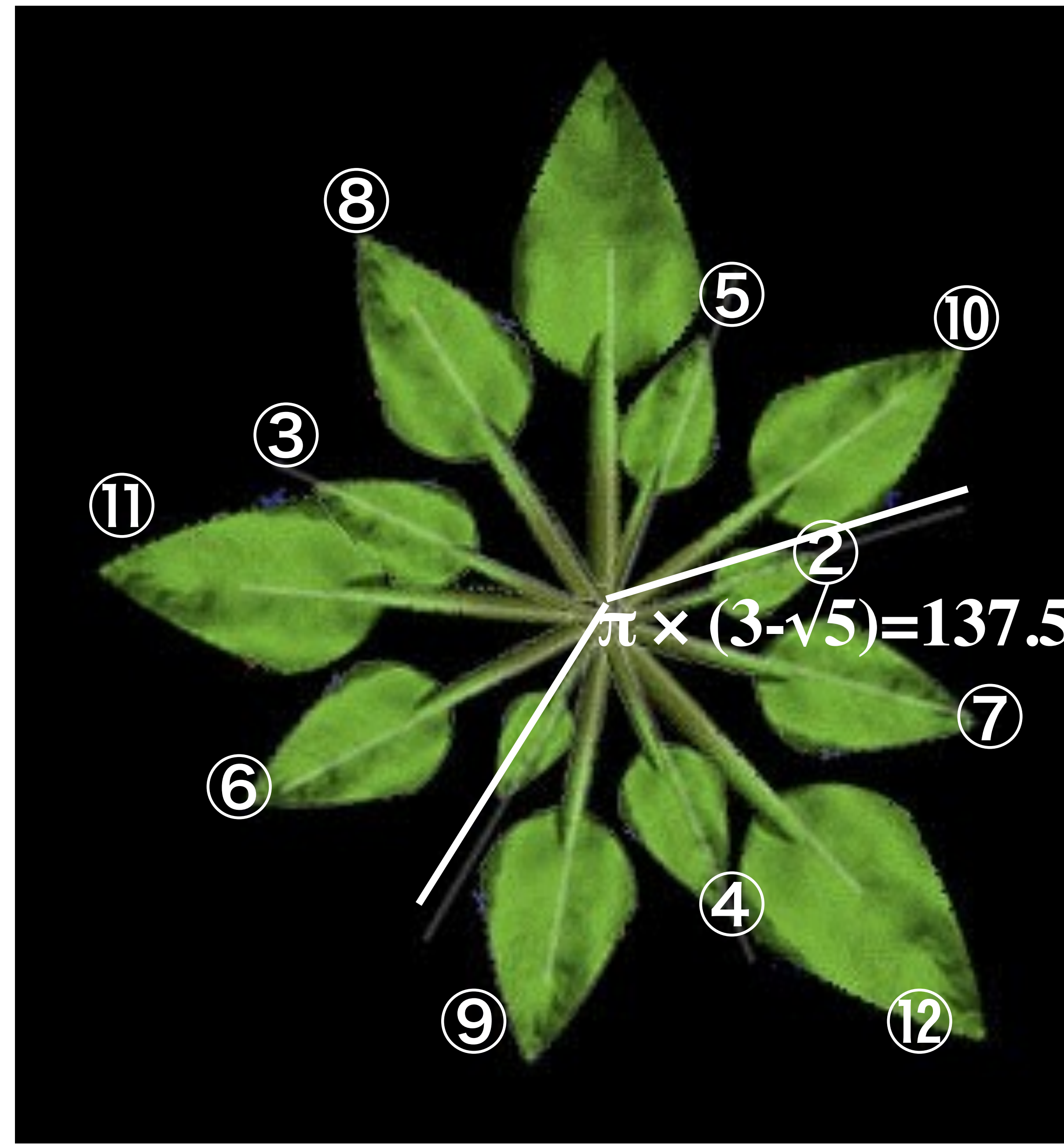
An irrational angle to π can approximate an arbitrary angle rapidly.



Density implies rapid cover

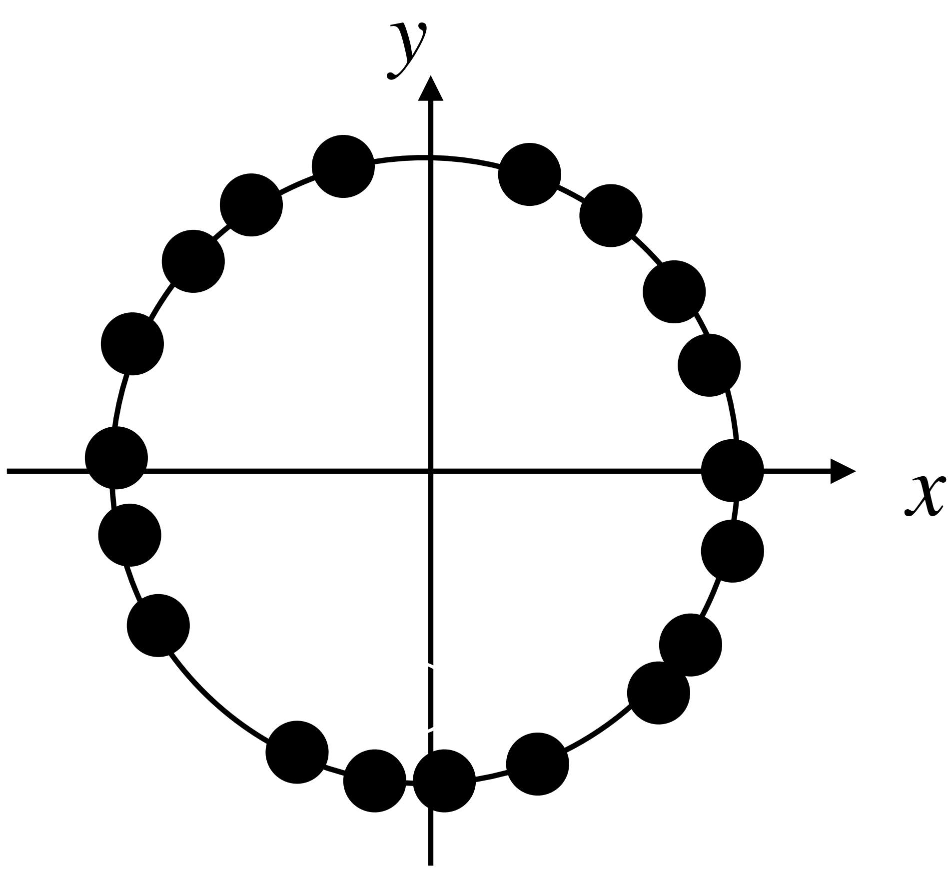


$$(3-\sqrt{5})\pi$$



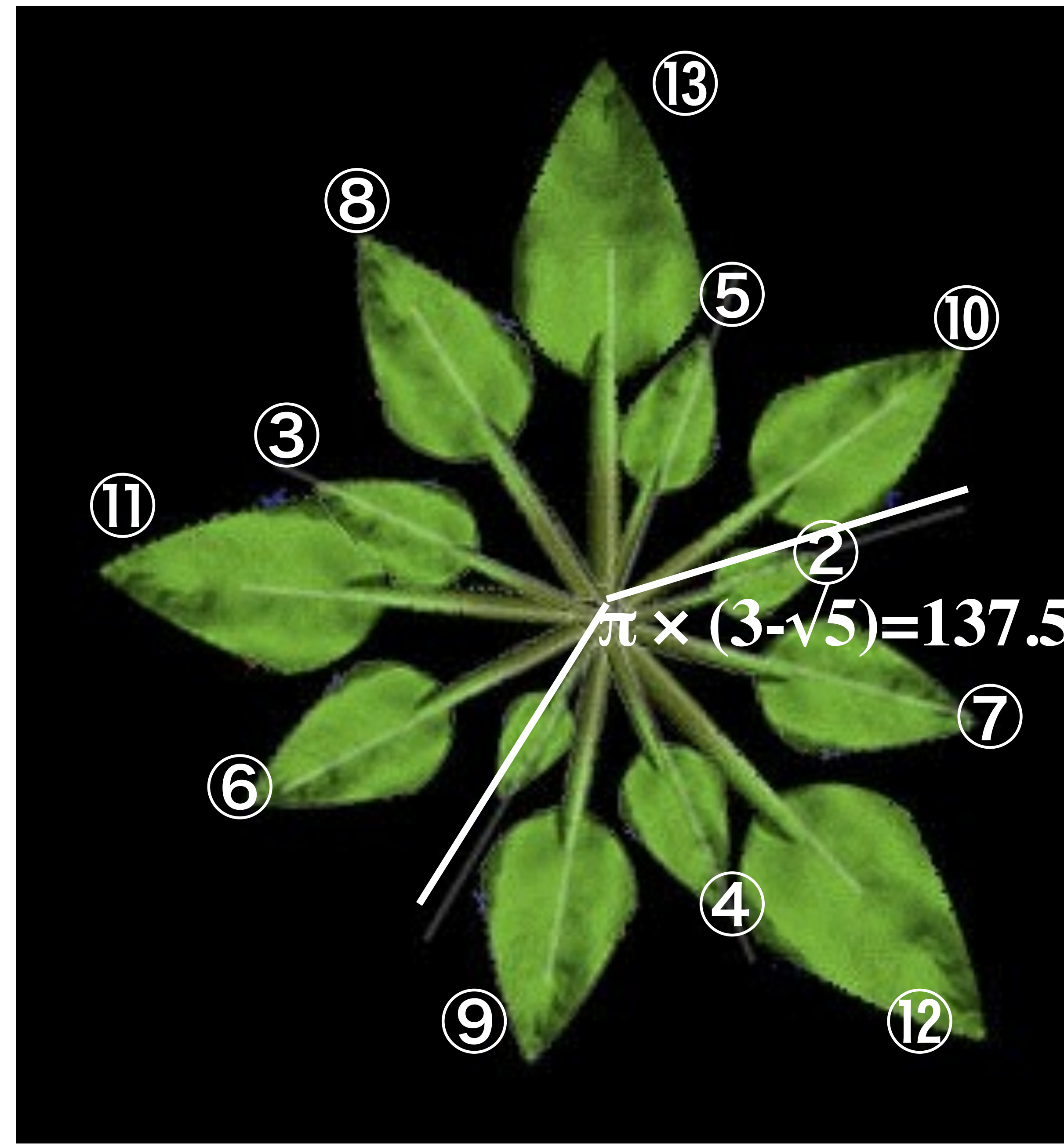
An irrational angle to π can approximate an arbitrary angle rapidly.

Density implies rapid cover



$$(3-\sqrt{5})\pi$$

An irrational angle to π can approximate an arbitrary angle rapidly.



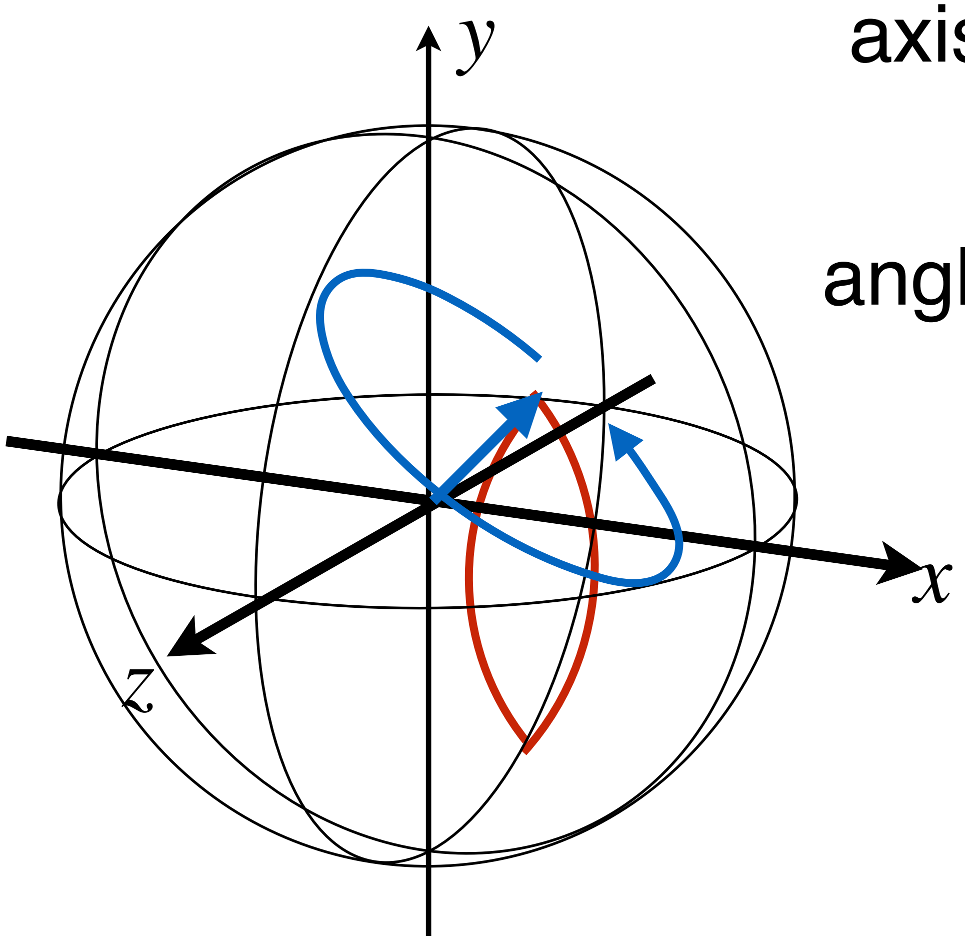
π/8 gate is enough

π/8 gate : $T = e^{-i(\pi/8)Z}$

$$THTH = \cos^2\frac{\pi}{8} I - i\left[\cos\frac{\pi}{8}(X+Z) + \sin\frac{\pi}{8} Y\right]\sin\frac{\pi}{8}$$

axis: $\left(\cos\frac{\pi}{8}, \sin\frac{\pi}{8}, \cos\frac{\pi}{8}\right)$

angle: $\theta = 2\arccos[\cos^2(\pi/8)]$



$\{H, T\}$ can approximate an arbitrary *SU*(2) efficiently.

Solovay-Kitaev algorithm

unitary Solovay-Kitaev(U,n){

if (n==0){ return basic approximation of U}

else{

U_{n-1}=Solovay-Kitaev(U,$n-1$);

V,W s.t. $VWV^\dagger W^\dagger = UU^\dagger_{n-1}$;

V_{n-1} = Solovay-Kitaev(V,$n-1$);

W_{n-1} = Solovay-Kitaev(V,$n-1$);

}

return $V_{n-1}W_{n-1}V_{n-1}^\dagger W_{n-1}^\dagger U_{n-1}$;

}

→ $O(\log^c(1/\epsilon))$

Dawson-Nielsen, QIC **6**, 81 (2006)

Multi-qubit system

Multi-qubit system

- Tensor product space:

$$|0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{array}{c} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \end{array} \begin{array}{l} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array}$$

Kronecker's product

- Product state:

$$|\psi_a\rangle \otimes |\psi_b\rangle$$

- Entangled state:

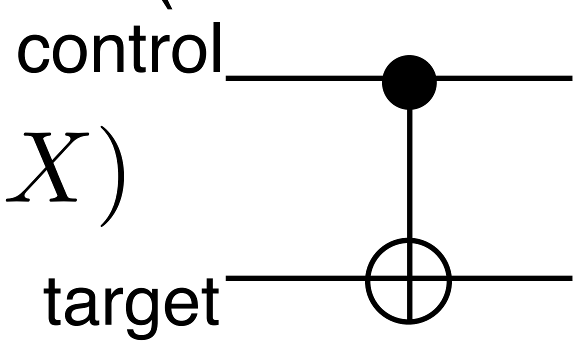
$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Multi-qubit gates

- Two-qubit gates：

CNOT (controlled NOT)

$\Lambda(X)$



quantum version of XOR

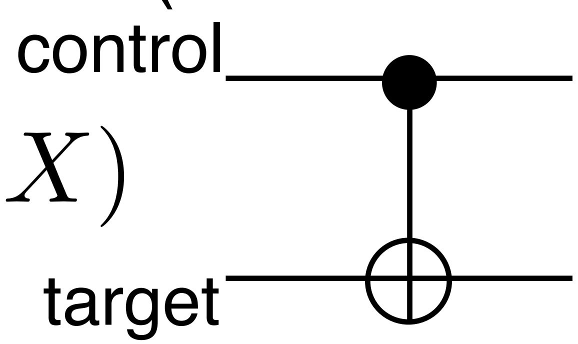
$$\begin{array}{cc} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & \begin{array}{c} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \end{array} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$$

Multi-qubit gates

- Two-qubit gates：

CNOT (controlled NOT)

$\Lambda(X)$

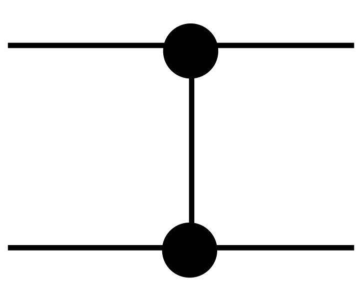


quantum version of XOR

$$\begin{array}{cc} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \end{array} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$$

CZ (controlled Z)

$\Lambda(Z)$



$$\begin{array}{cc} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} & \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \end{array} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z$$

Universal quantum computation

- Solovay-Kitaev algorithm : $\{H, T\} \rightarrow$ an arbitrary single-qubit gate
- CNOT + single-qubit gate → an arbitrary *n*-qubit unitary gate

universal set $\{\Lambda(X), H, T\}$

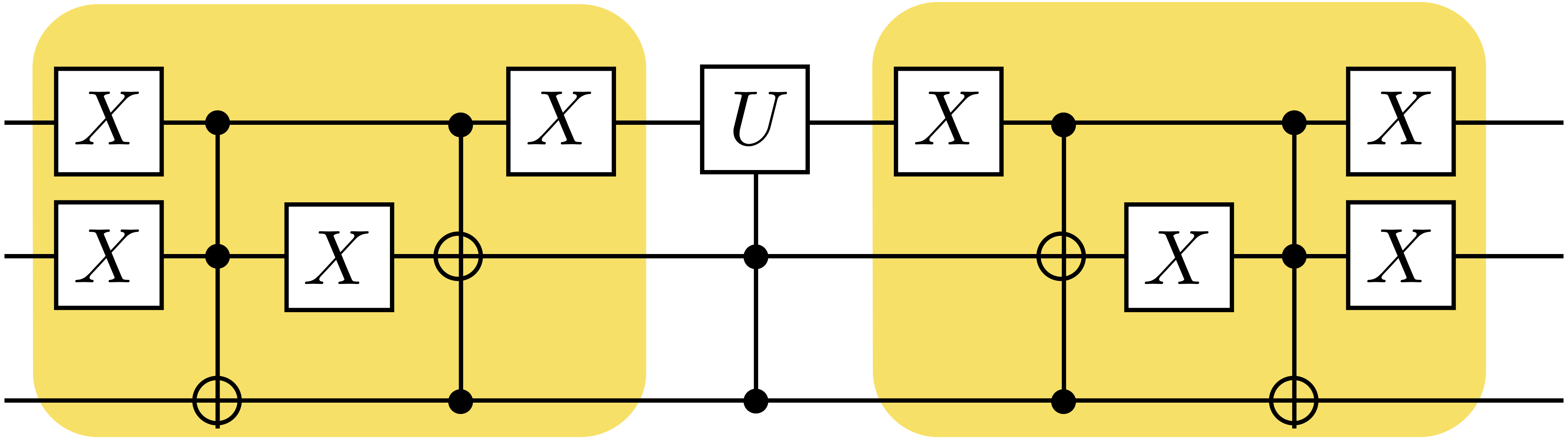
Universal quantum computation

- Solovay-Kitaev algorithm： $\{H, T\} \rightarrow$ an arbitrary single-qubit gate

- CNOT + single-qubit gate → an arbitrary *n*-qubit unitary gate

universal set $\{\Lambda(X), H, T\}$

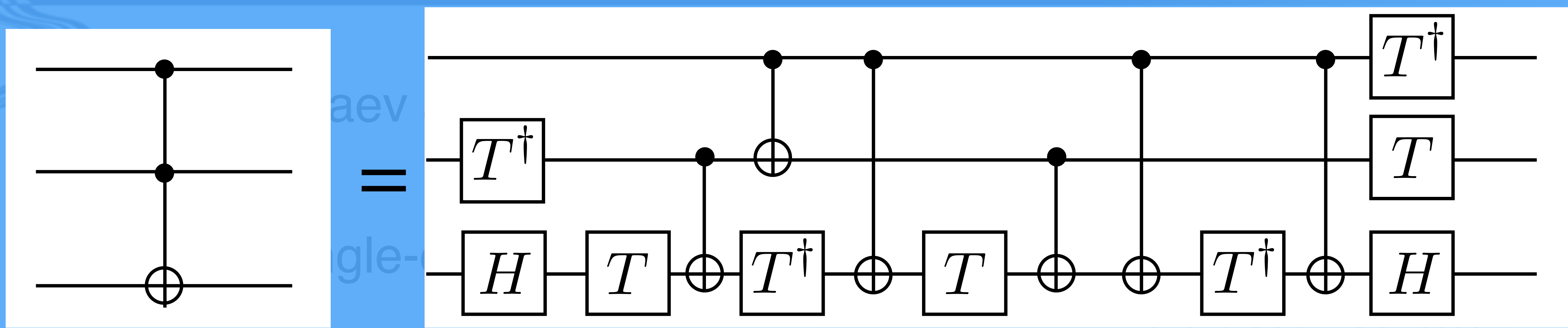
For example a rotation for $\{|000\rangle, |111\rangle\}$:



000→ 011
111→ 111

Universal quantum computation

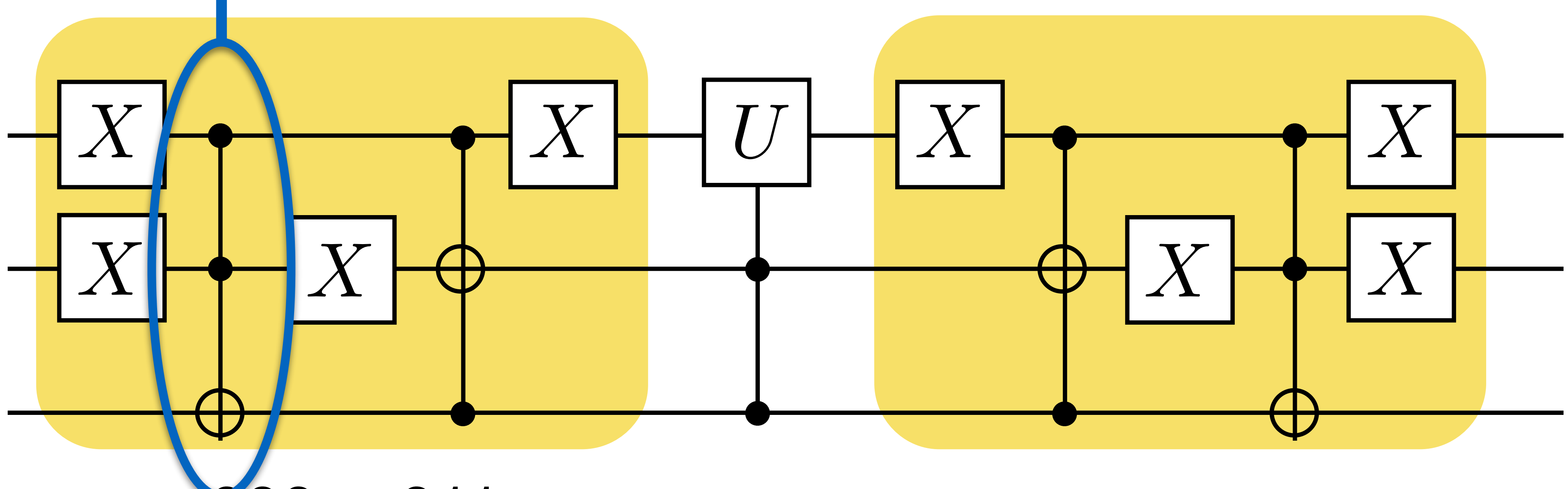
Toffoli gate (quantum version of NAND gate)



universal set $\{\Lambda(X), H, T\}$

$T = e^{-(\pi/8)iZ}$

For example a rotation for $\{|000\rangle, |111\rangle\}$:



000→ 011
111→ 111

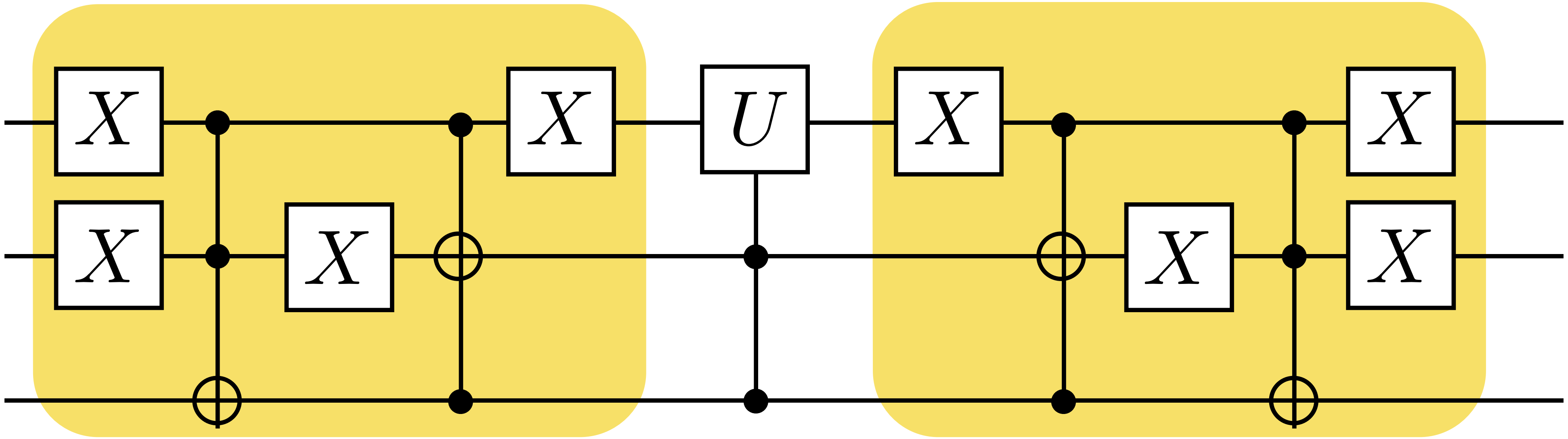
Universal quantum computation

- Solovay-Kitaev algorithm : $\{H, T\} \rightarrow$ an arbitrary single-qubit gate

- CNOT + single-qubit gate → an arbitrary *n*-qubit unitary gate

universal set $\{\Lambda(X), H, T\}$

For example a rotation for $\{|000\rangle, |111\rangle\}$:

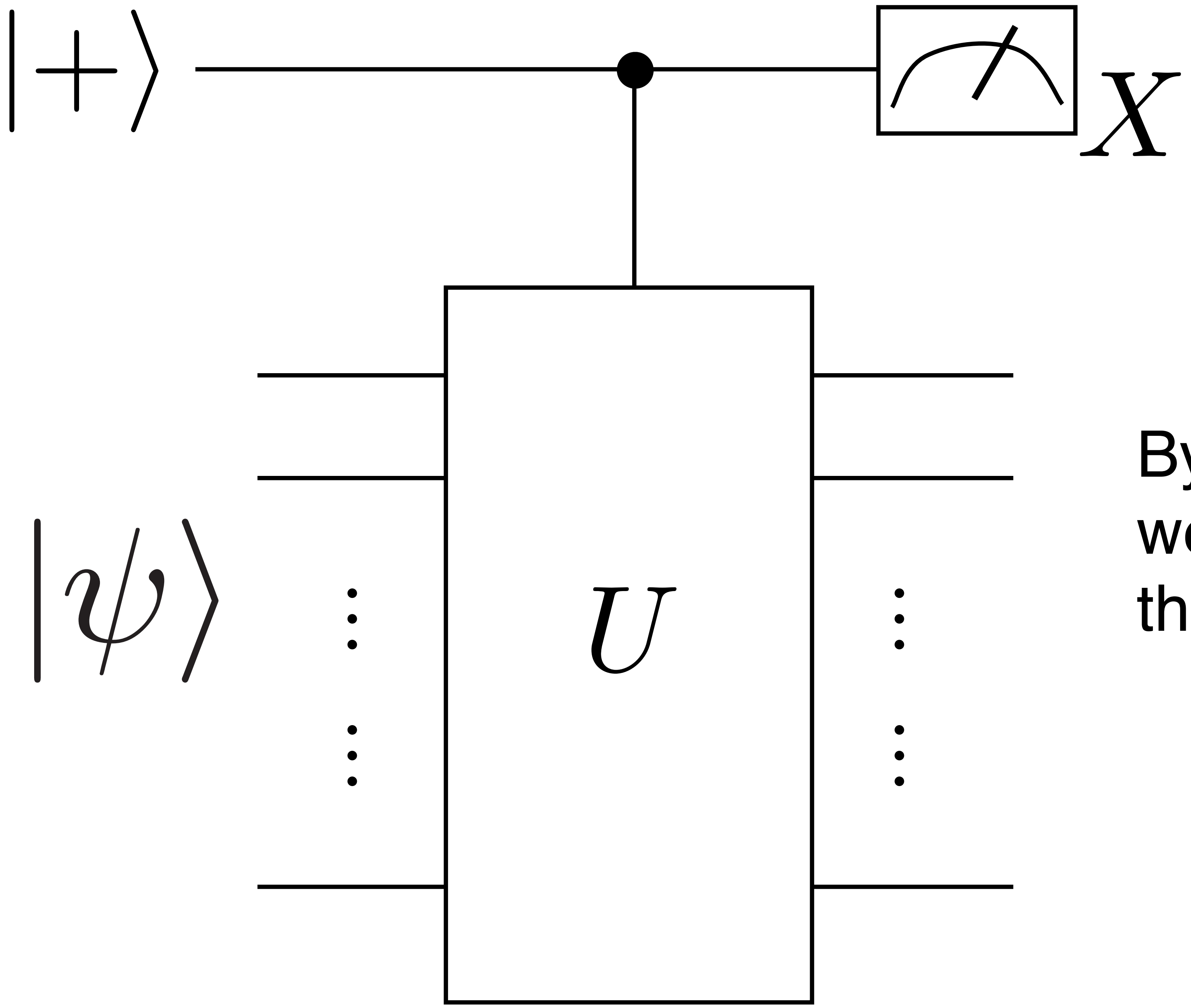


000→ 011
111→ 111

How quantum algorithms work

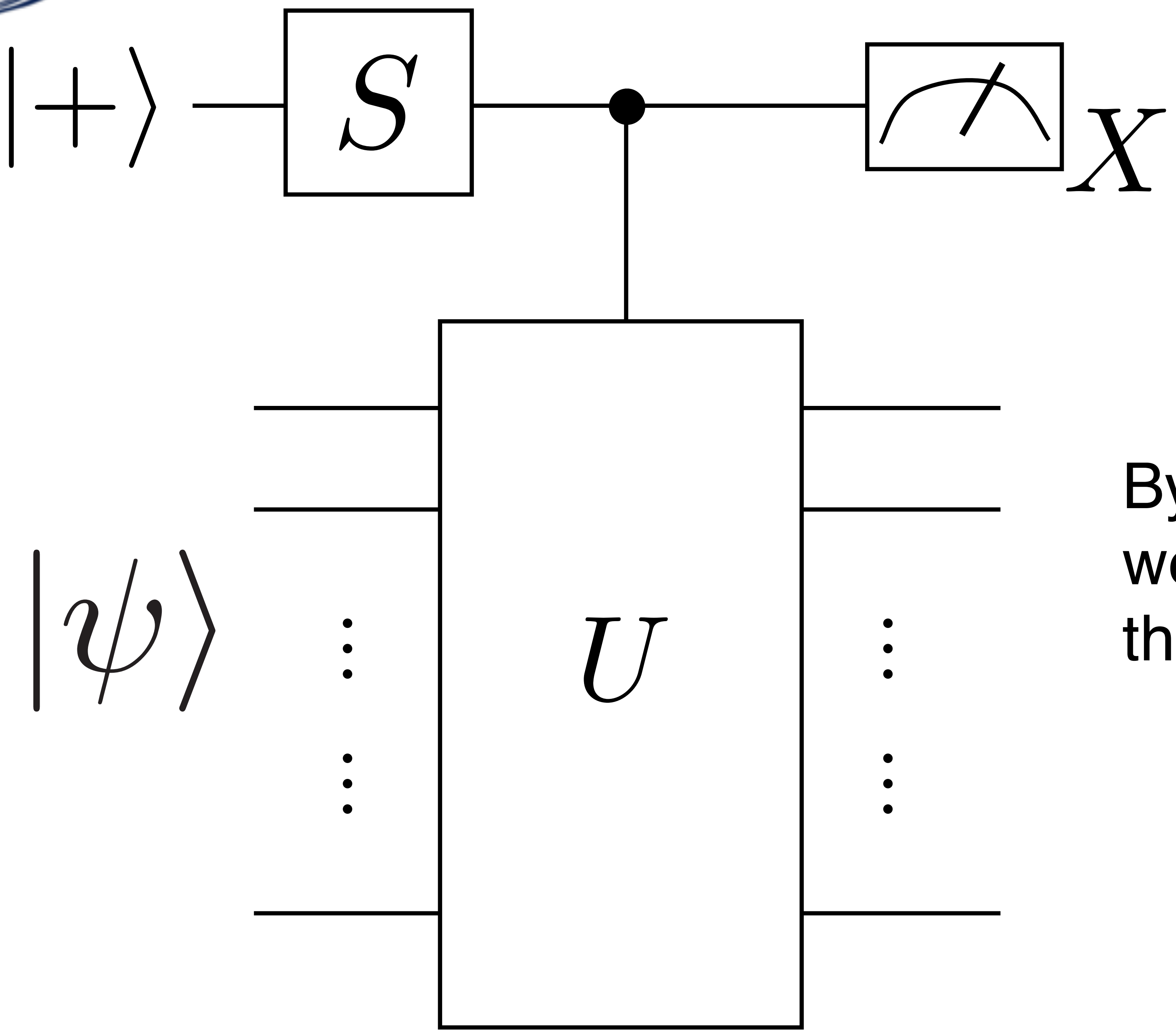
Hadamard test

$$p_+ = \frac{1}{2}\left(1 + \mathrm{Re}\langle\psi|U|\psi\rangle\right)$$



By repeating the Hadamard test, we can obtain a matrix element of the unitary *U*.

Hadamard test

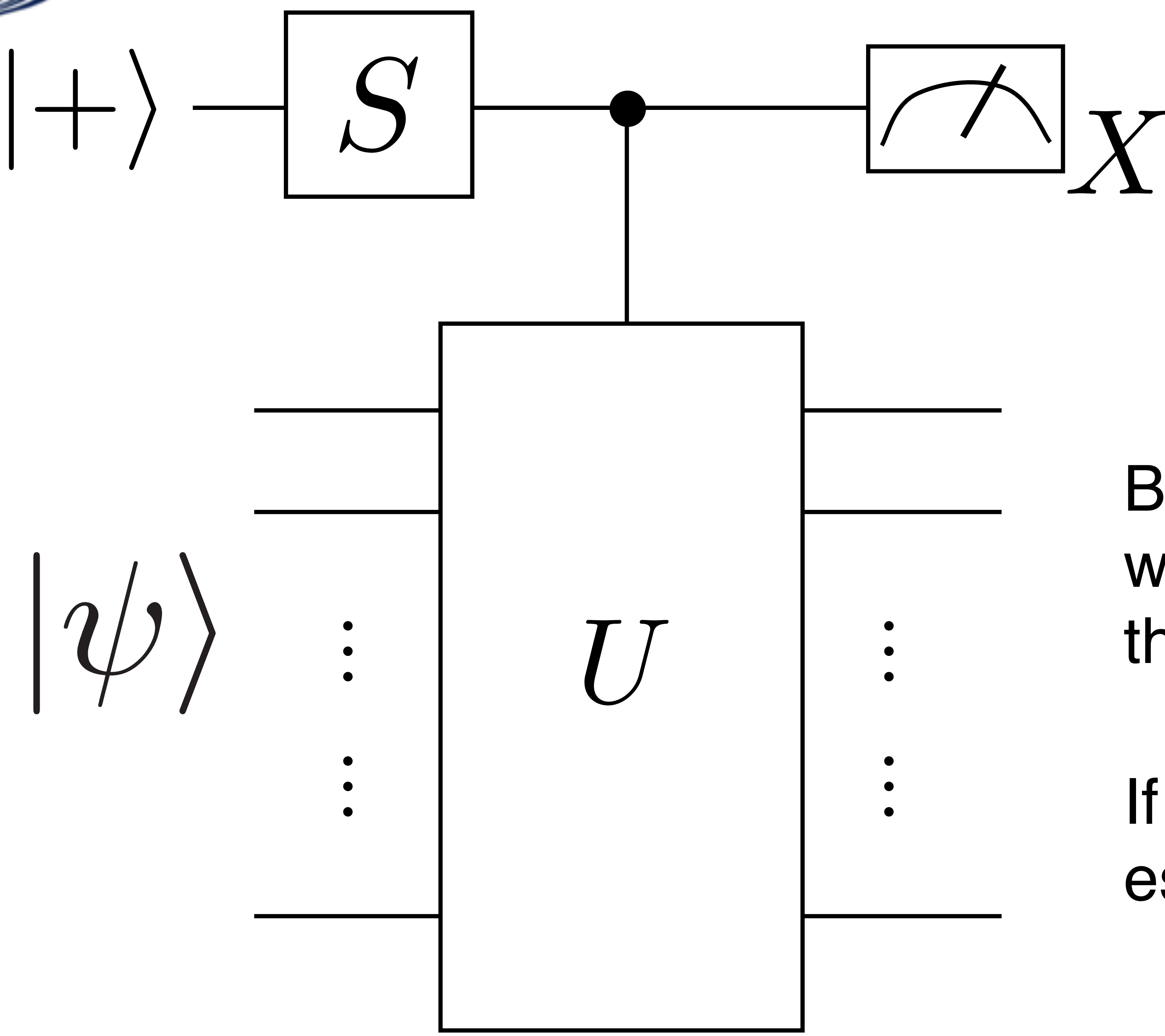


$$p_+ = \frac{1}{2}\left(1 + \mathrm{Re}\langle\psi|U|\psi\rangle\right)$$

$$p_+ = \frac{1}{2}\left(1 + \mathrm{Im}\langle\psi|U|\psi\rangle\right)$$

By repeating the Hadamard test, we can obtain a matrix element of the unitary *U*.

Hadamard test



$$p_+ = \frac{1}{2}\left(1 + \mathrm{Re}\langle\psi|U|\psi\rangle\right)$$

$$p_+ = \frac{1}{2}\left(1 + \mathrm{Im}\langle\psi|U|\psi\rangle\right)$$

By repeating the Hadamard test, we can obtain a matrix element of the unitary *U*.

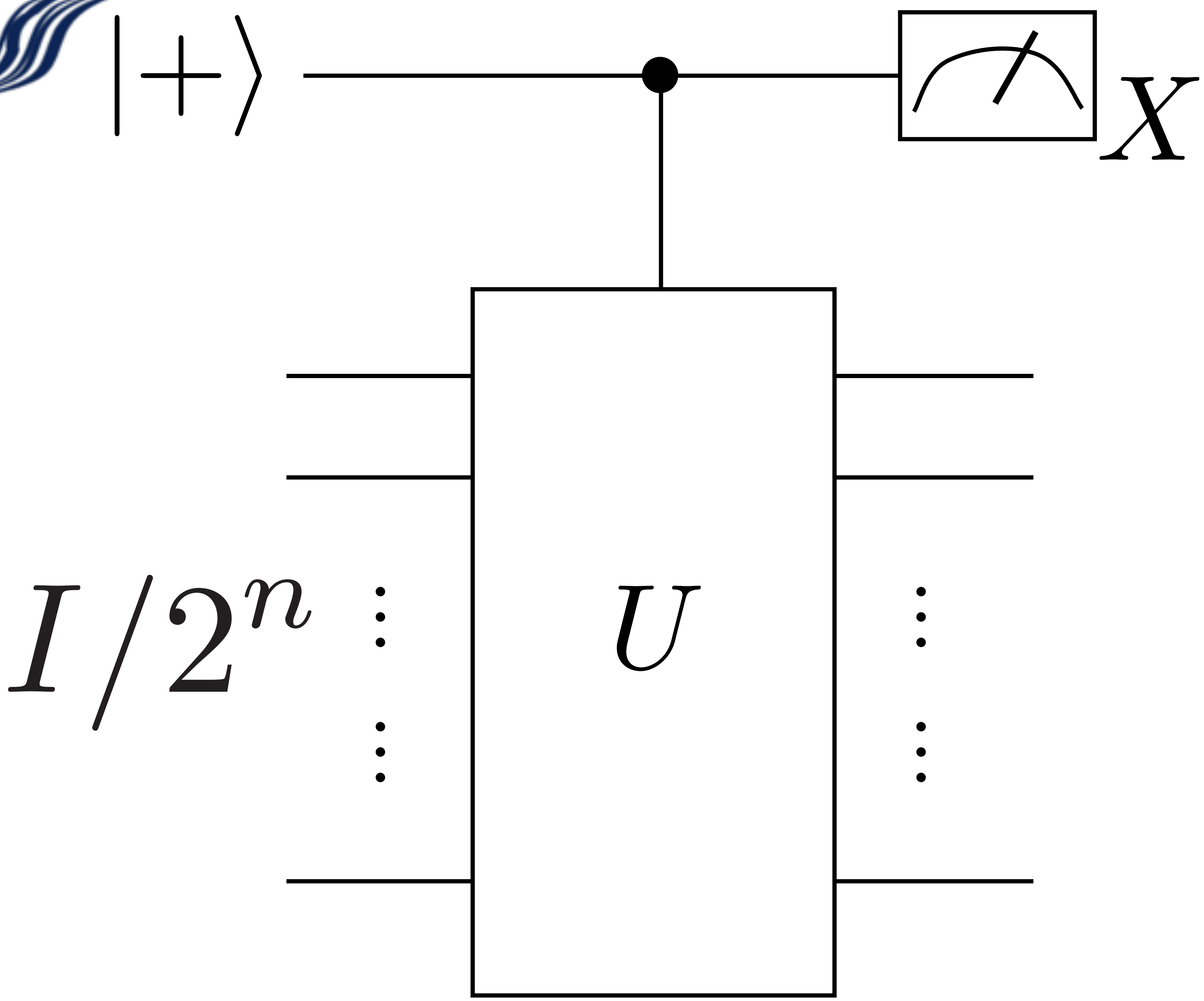
If $|\psi\rangle$ is an eigenstate of *U*, we can estimate **an eigenvalue** of *U*.

E_i $\qquad e^{-iHt}$

DQC1

deterministic quantum computation with one-clean qubit

[Knill-Laflamme, PRL 81, 5672 (1998) ; G. Passante et al., PRL **103**, 250501 (2009)]



$$\longrightarrow \mathrm{Tr}[U]/2^n$$

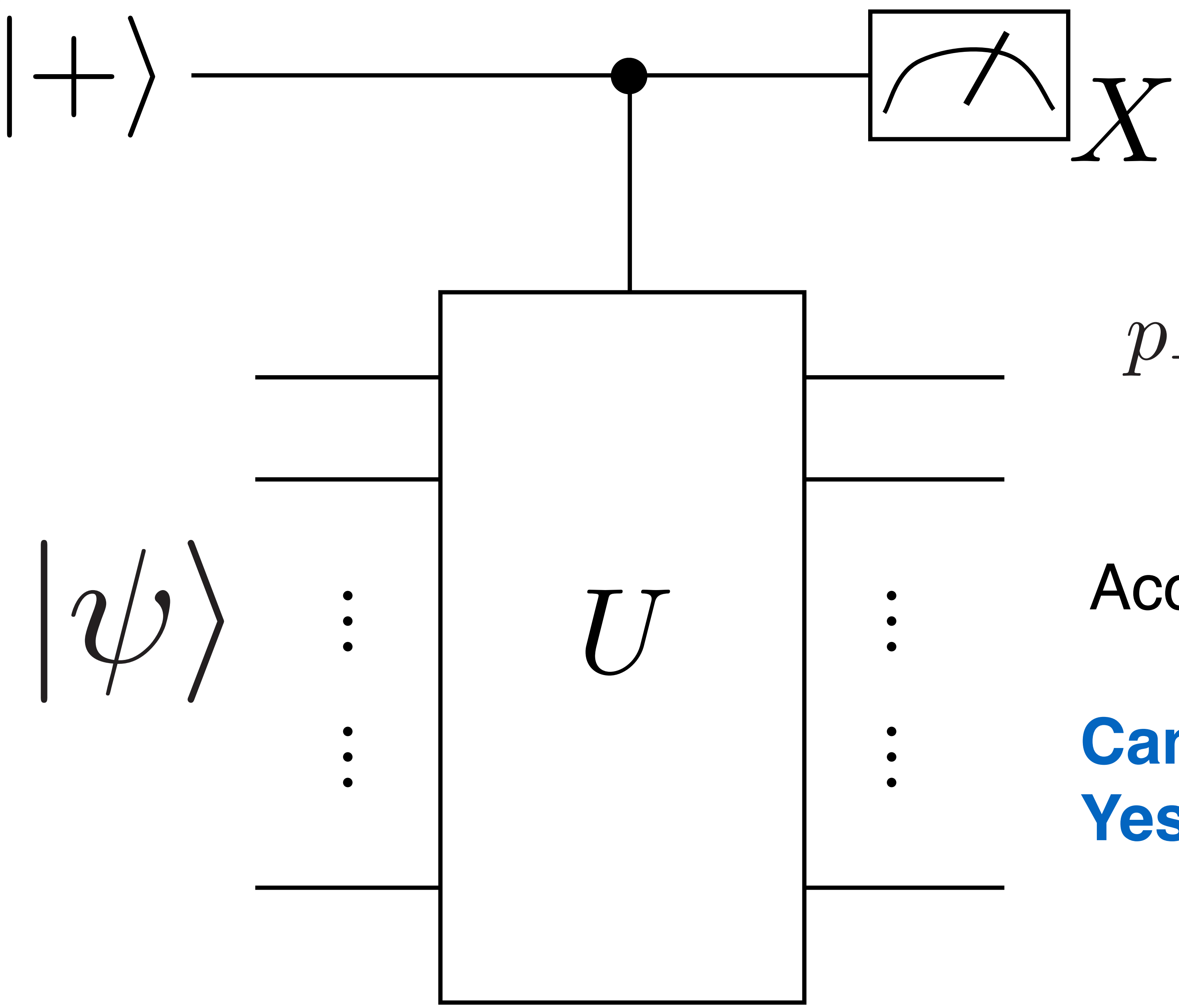
Not universal but still useful….

fidelity decay, Jones polynomial, HOMFLY polynomial….

[Poulin *et al*., PRL **92**, 177906 (2004). Shor-Jordan, QIC **8**, 681 (2008)]

[Morimae-KF-Fitzsimons, PRL '14; KF et al., arXiv:1509.07276]

Hadamard test



$$p_+ = \frac{1}{2}(1 + \mathrm{Re}\langle\psi|U|\psi\rangle)$$

Accuracy of estimation →1/poly(N)

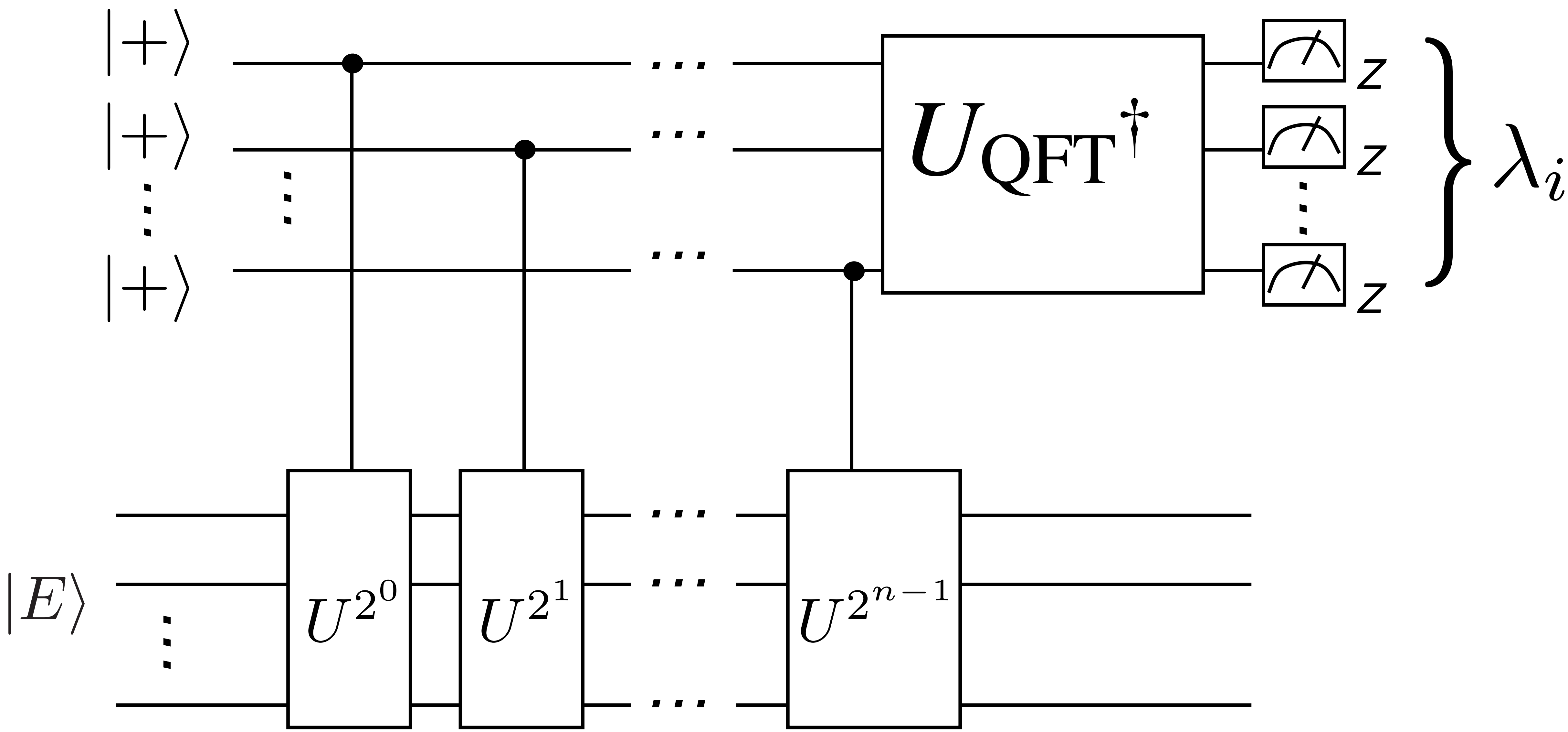
Can we improve the accuracy?
Yes, if *U* has a special property!

Kitaev's phase estimation

Suppose the eigenvalue is $e^{(2\pi i)0.j_1 j_2 \ldots j_n}$, where

$$0.j_1 j_2 \ldots j_n = \sum_k^n j_k (1/2)^k.$$

binary fraction

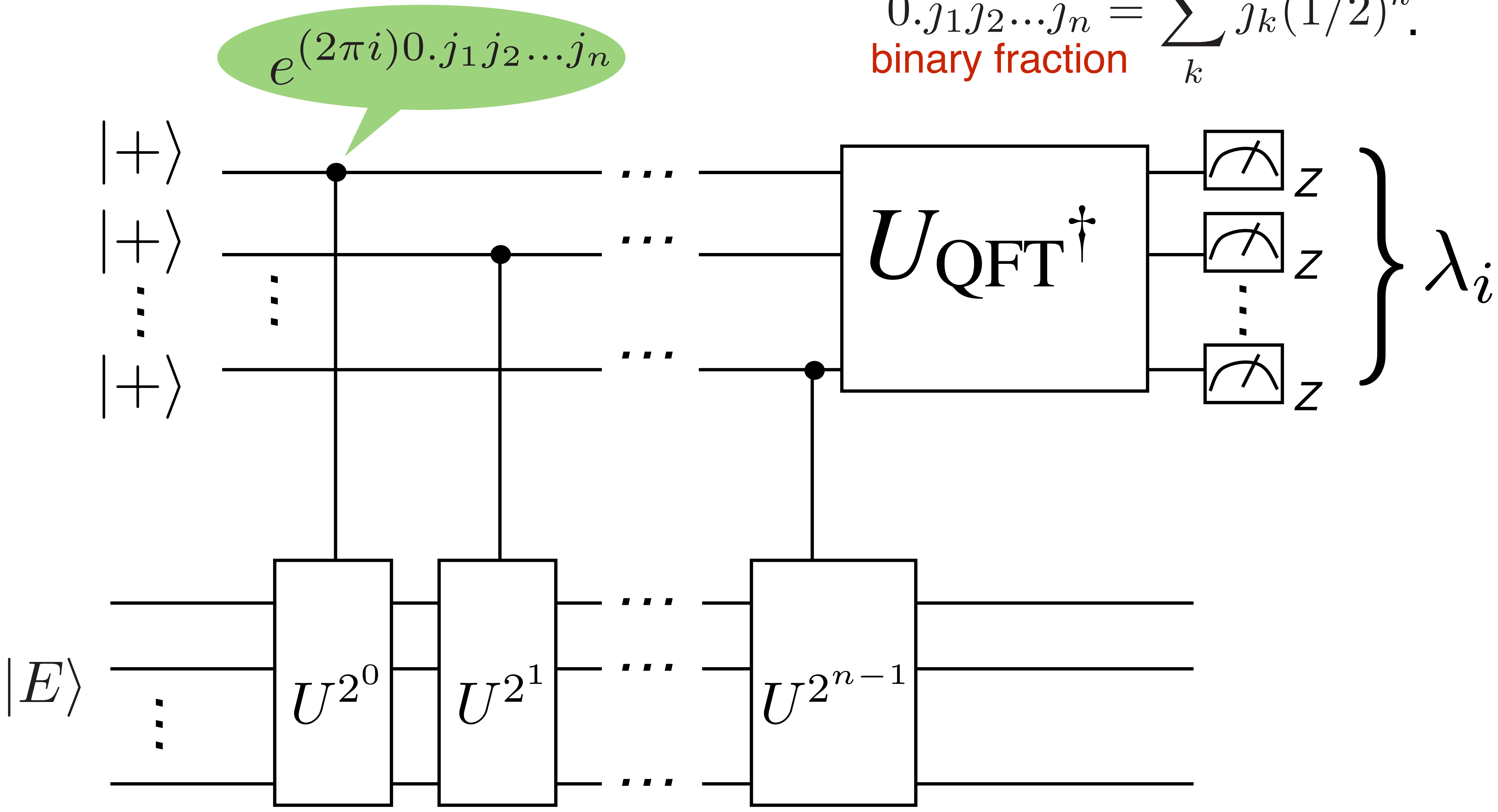


Kitaev's phase estimation

Suppose the eigenvalue is $e^{(2\pi i)0.j_1 j_2 \ldots j_n}$, where

$$0.j_1 j_2 \ldots j_n = \sum_k^n j_k (1/2)^k.$$

binary fraction

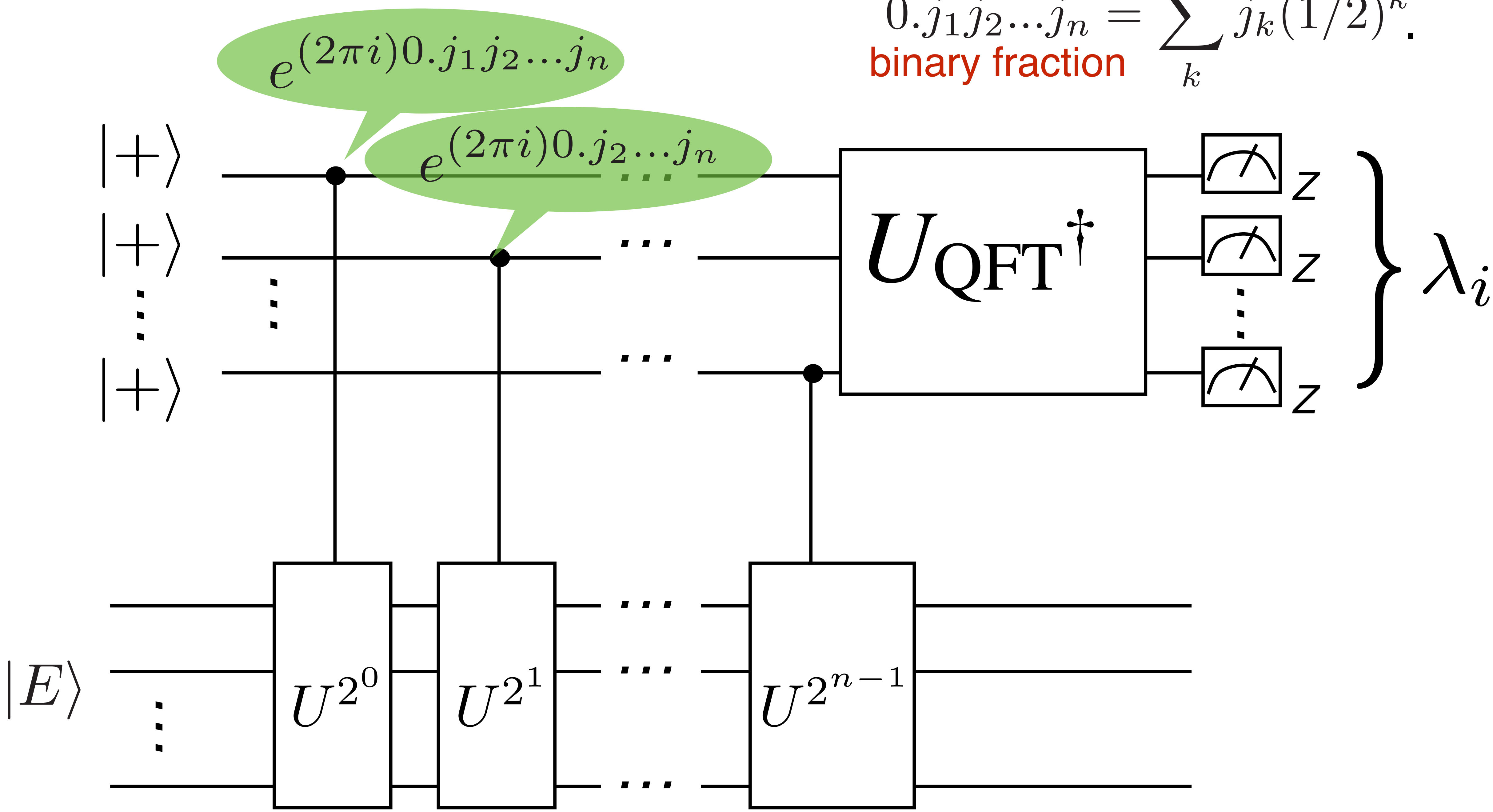


Kitaev’s phase estimation

Suppose the eigenvalue is $e^{(2\pi i)0.j_1 j_2 \dots j_n}$, where

$$0.j_1 j_2 \dots j_n = \sum_k^n j_k (1/2)^k.$$

binary fraction

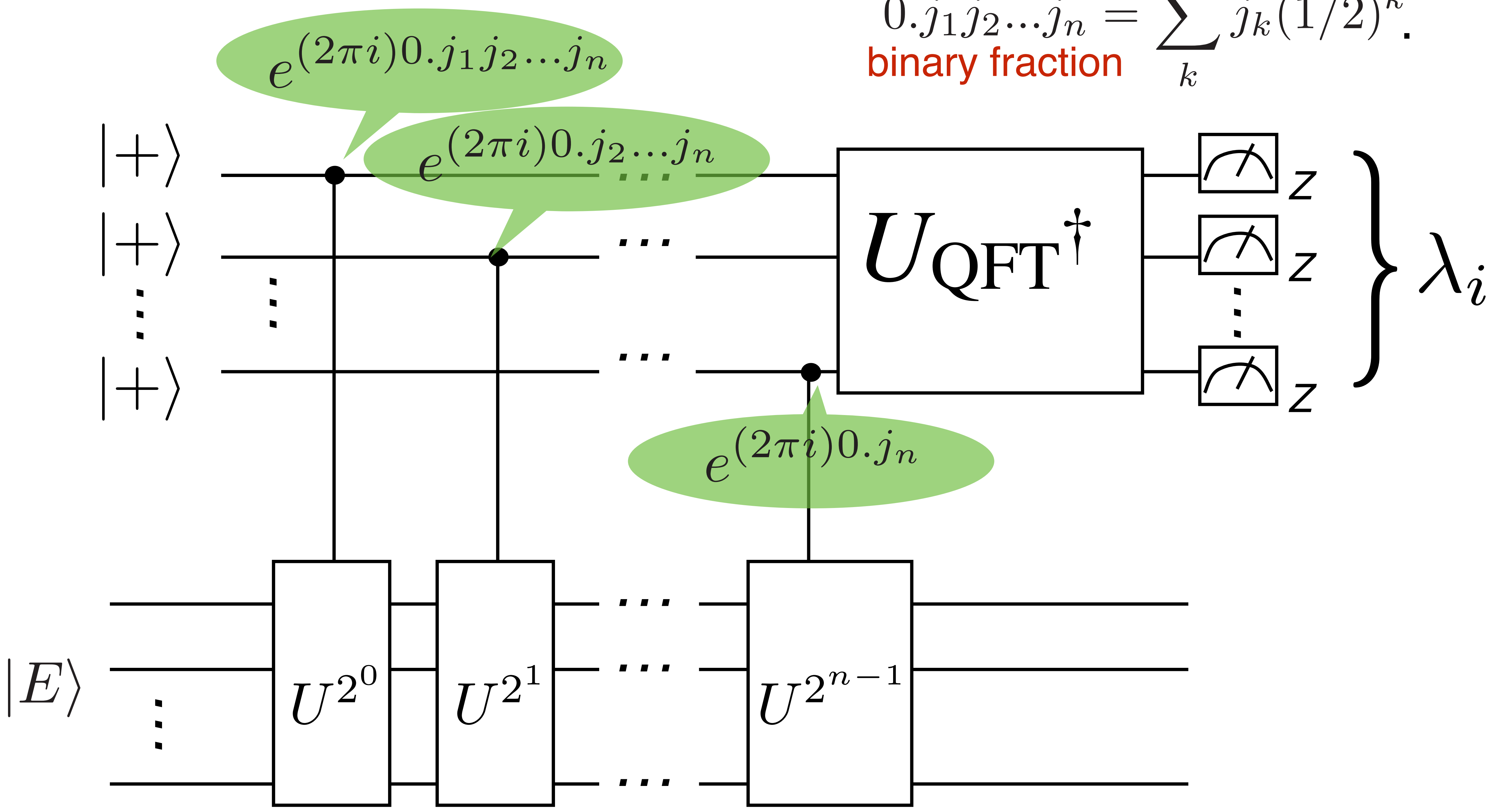


Kitaev's phase estimation

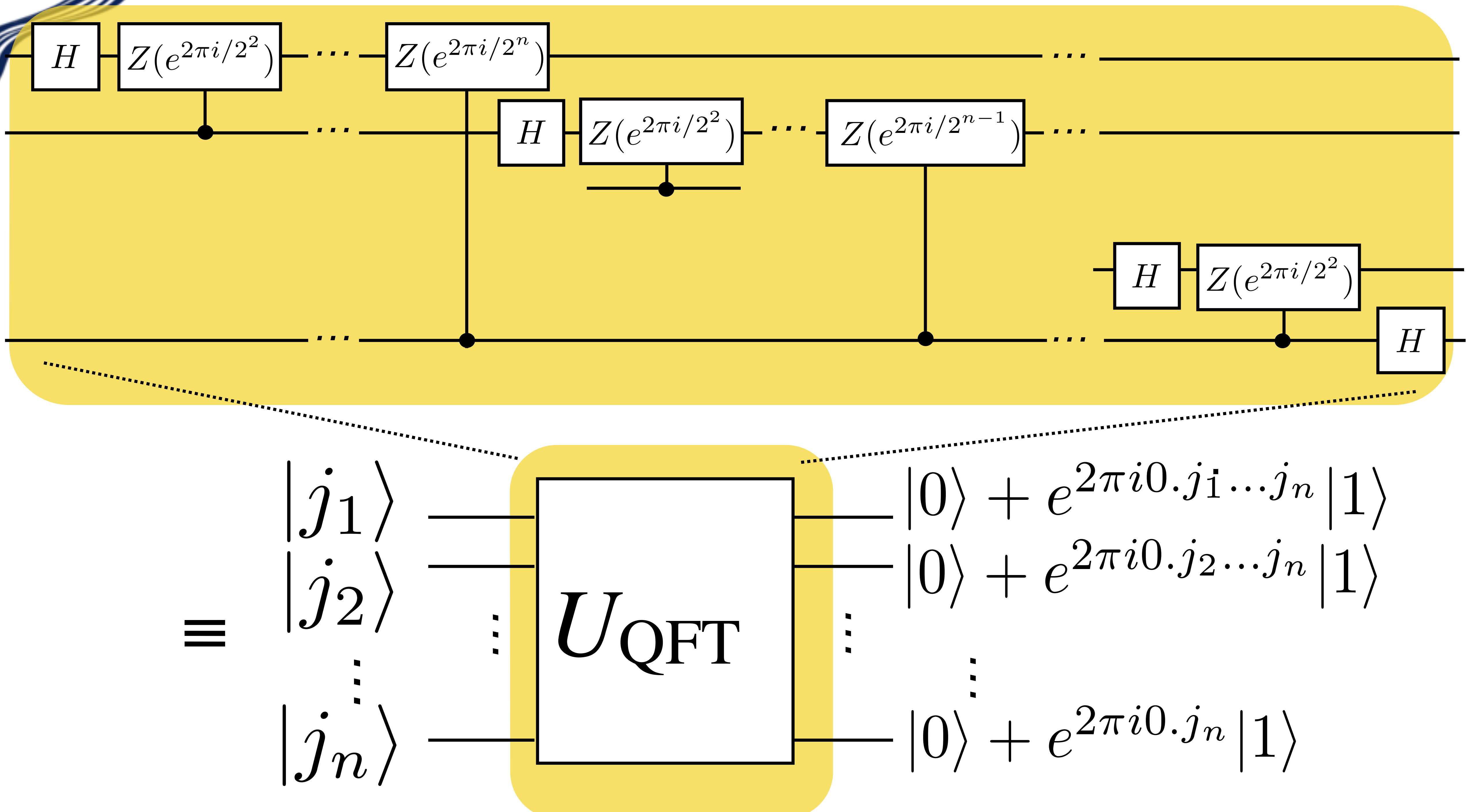
Suppose the eigenvalue is $e^{(2\pi i)0.j_1 j_2 \ldots j_n}$, where

$$0.j_1 j_2 \ldots j_n = \sum_k^n j_k (1/2)^k.$$

binary fraction



Quantum Fourier transformation

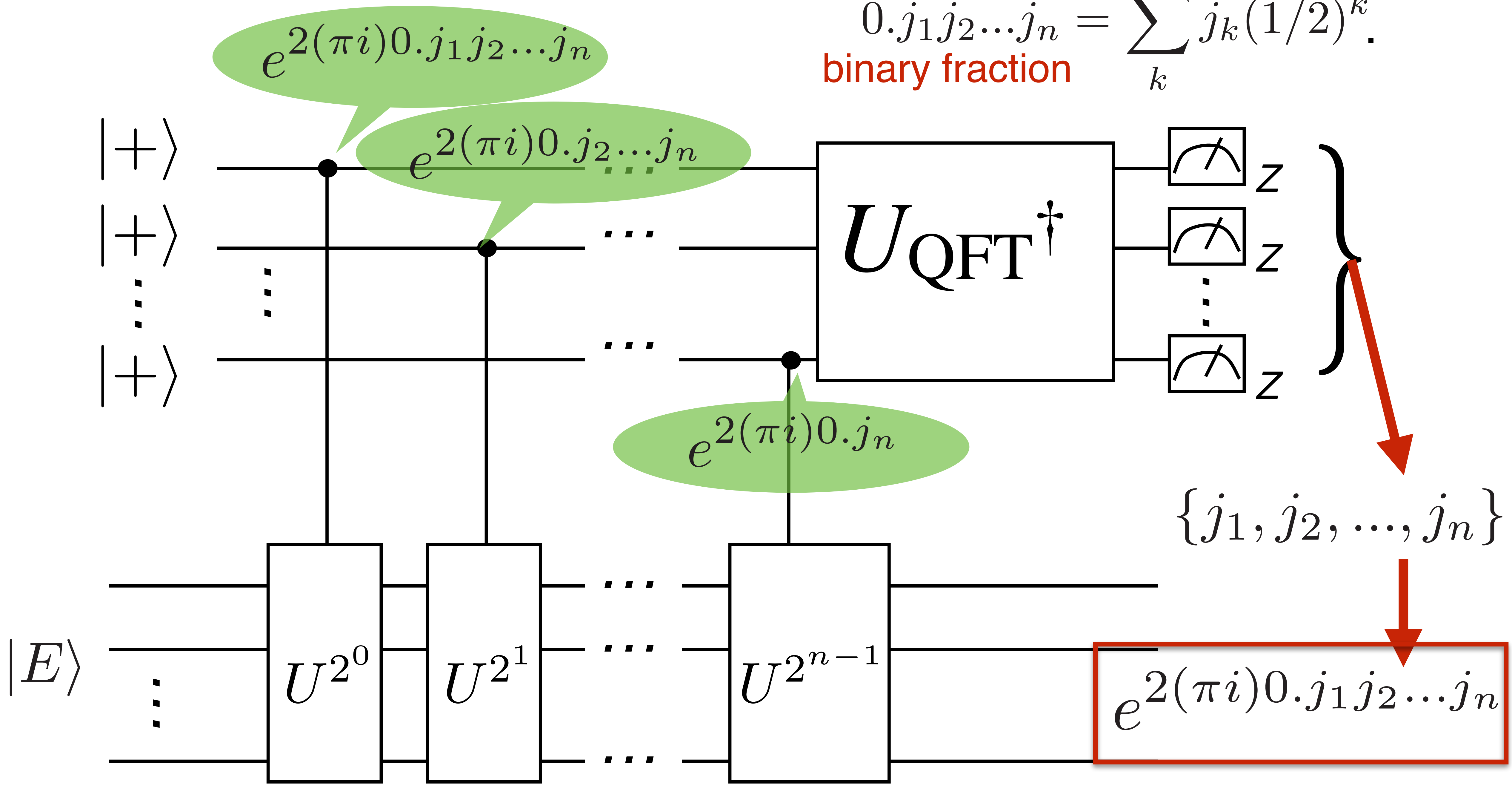


Kitaev's phase estimation

Suppose the eigenvalue is $e^{2(\pi i)0.j_1 j_2 \ldots j_n}$, where

$$0.j_1 j_2 \ldots j_n = \sum_k^n j_k (1/2)^k.$$

binary fraction



Shor's factoring algorithm

Let N be an integer that we want to factorize.

$N = 15$

Shor's factoring algorithm

Let N be an integer that we want to factorize.

Choose a co-prime x, and estimate an eigenvalue of the modular exponentiation:

$$U_x = \sum_y |xy \bmod N\rangle\langle y|$$

$N = 15$

$x = 7$

Shor's factoring algorithm

Let N be an integer that we want to factorize.

Choose a co-prime x, and estimate an eigenvalue of the modular exponentiation:

$$U_x = \sum_y |xy \bmod N\rangle\langle y|$$

Fact 1: Let r be an order such that $x^r = 1 \bmod N$.

Then we have $U_x|u_s\rangle = e^{2\pi i(s/r)}|u_s\rangle$

Kitaev's phase estimation

$N = 15$

$x = 7$

$7^4 = 1$

$r = 4$

Shor's factoring algorithm

Let N be an integer that we want to factorize.

Choose a co-prime x, and estimate an eigenvalue of the modular exponentiation:

$$U_x = \sum_y |xy \bmod N\rangle\langle y|$$

Fact 1: Let r be an order such that $x^r = 1 \bmod N$.

Then we have $U_x|u_s\rangle = e^{2\pi i(s/r)}|u_s\rangle$

Kitaev's phase estimation

Fact 2: r is even with a high probability.

$N = 15$

$x = 7$

$7^4 = 1$

$r = 4$

Shor's factoring algorithm

Let N be an integer that we want to factorize.

Choose a co-prime x, and estimate an eigenvalue of the modular exponentiation:

$$U_x = \sum_y |xy \bmod N\rangle\langle y|$$

Fact 1: Let r be an order such that $x^r = 1 \bmod N$.

Then we have $U_x|u_s\rangle = e^{2\pi i(s/r)}|u_s\rangle$

Kitaev's phase estimation

Fact 2: r is even with a high probability.

Finally we obtain $(x^{r/2}-1)(x^{r/2}+1) = 0 \bmod N$.

GCD of $(x^{r/2} \pm 1)$ and N is the factor of N!

$N = 15$

$x = 7$

$7^4 = 1$

$r = 4$

$(7^2-1)(7^2+1)=0$

48 50

How complex quantum states are described efficiently

Stabilizer formalism

- *n*-qubit system

Exponentially many parameters!

$$|\psi_n\rangle = \sum_{s_1, s_2, \ldots, s_n} c_{s_1 s_2 \ldots s_n} |s_1 s_2 \ldots s_n\rangle$$

$$|s_1 s_2 \ldots s_n\rangle \equiv |s_1\rangle \otimes |s_2\rangle \otimes \cdots \otimes |s_n\rangle$$

Stabilizer formalism

- *n*-qubit system

Exponentially many parameters!

$$|\psi_n\rangle = \sum_{s_1, s_2, \ldots, s_n} c_{s_1 s_2 \ldots s_n} |s_1 s_2 \ldots s_n\rangle$$

$$|s_1 s_2 \ldots s_n\rangle \equiv |s_1\rangle \otimes |s_2\rangle \otimes \cdots \otimes |s_n\rangle$$

→ *Stabilizer formalism*

Resource states for MBQC, Quantum error correction codes

(also utilized in condensed matter physics, and particle physics)

[D. Gottesman, Ph.D. thesis, California Institute of Technology (1997); arXiv:quant- ph/9705052.]

Stabilizer group

- *n*-qubit Pauli group:

$$\boxed{\{\pm 1, \pm i\} \times \{I, X, Y, Z\}^{\otimes n} \in \mathcal{P}_n}$$

e.g. $\{\pm 1, \pm i\} \times \{II, IX, IY, IZ, ..., ZZ\}$

16 elements

$$\boxed{A \otimes B = AB}$$

Stabilizer group

- n-qubit Pauli group:

$$\{\pm 1, \pm i\} \times \{I, X, Y, Z\}^{\otimes n} \in \mathcal{P}_n$$

e.g. $\{\pm 1, \pm i\} \times \{II, IX, IY, IZ, ..., ZZ\}$
16 elements

$$A \otimes B = AB$$

- Stabilizer group $\mathcal{S} = \{S_i\}$: hermitian and Abelian subgroup of the Pauli group

$$S_i \in \mathcal{P}, \quad S_i = S_i^\dagger, \quad [S_i, S_j] = 0$$

e.g. $\langle XX, ZZ \rangle = \{II, XX, ZZ, -YY\}$

stabilizer generator
= maximum independent subset

even overlap

(anti-comm.)×2 = comm.

Stabilizer state

- Stabilizer state

$$S_i|\Psi\rangle = |\Psi\rangle \text{ for all } S_i \in \mathcal{S}$$

  - Stabilizer group is Abelian and hence simultaneously diagonalized.
  - It is enough to check for all generators.

Stabilizer state

◆ Stabilizer state

$$\boxed{S_i|\Psi\rangle = |\Psi\rangle \text{ for all } S_i \in \mathcal{S}}$$

- Stabilizer group is Abelian and hence simultaneously diagonalized.
- It is enough to check for all generators.

example1:

Bell state

$$\mathcal{S}_1 = \langle XX, ZZ \rangle \longrightarrow (|00\rangle + |11\rangle)/\sqrt{2}$$

Stabilizer state

◆ Stabilizer state

$$S_i|\Psi\rangle = |\Psi\rangle \text{ for all } S_i \in \mathcal{S}$$

- Stabilizer group is Abelian and hence simultaneously diagonalized.
- It is enough to check for all generators.

example1:

Bell state

$$\mathcal{S}_1 = \langle XX, ZZ\rangle \longrightarrow (|00\rangle + |11\rangle)/\sqrt{2}$$

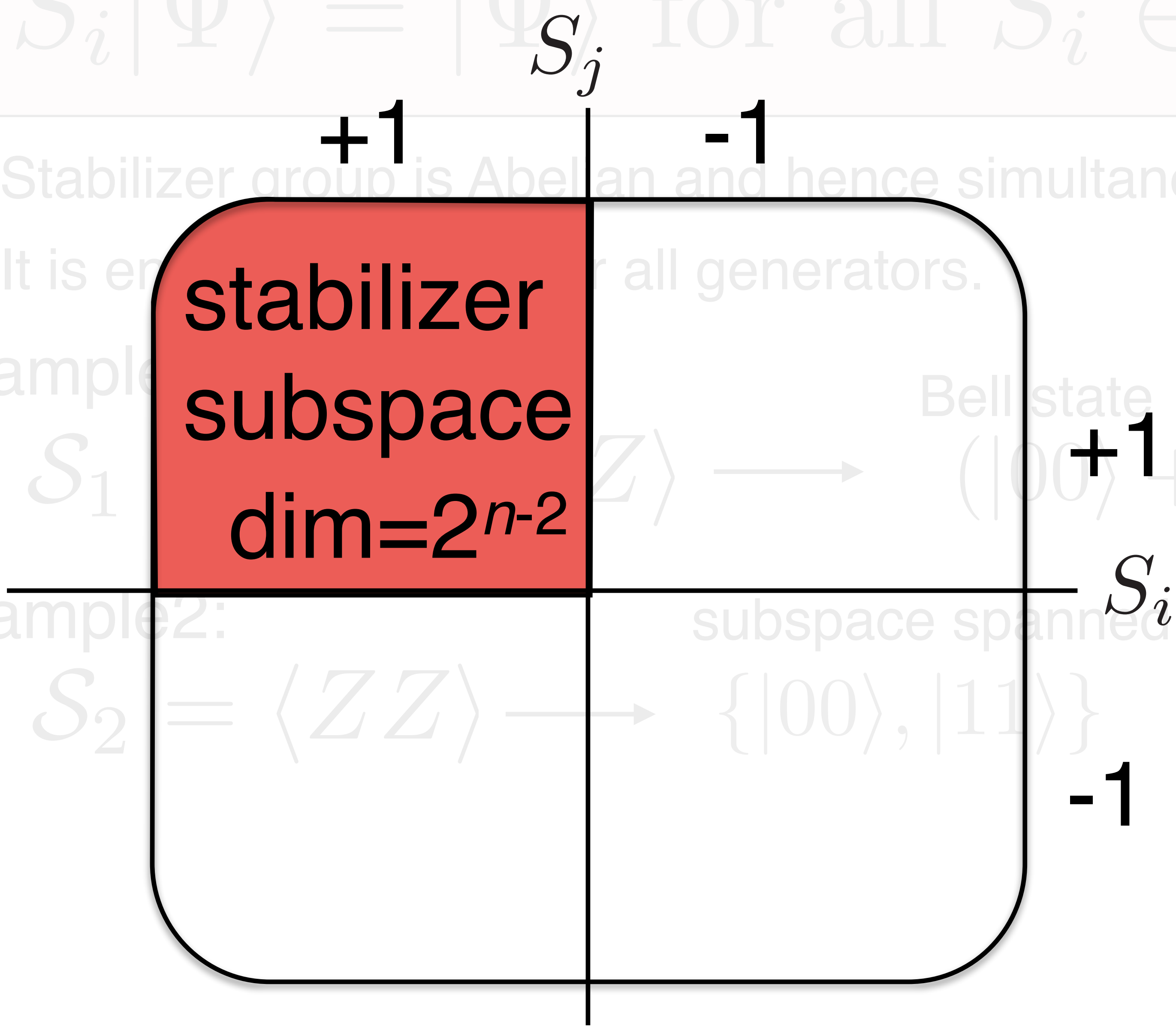
example2:

subspace spanned by

$$\mathcal{S}_2 = \langle ZZ\rangle \longrightarrow \{|00\rangle, |11\rangle\}$$

Stabilizer state

n qubits space → dim=2^n



Stabilizer state

- Stabilizer state

$$\boxed{S_i|\Psi\rangle = |\Psi\rangle \text{ for all } S_i \in \mathcal{S}}$$

- Stabilizer group is Abelian and hence simultaneously diagonalized.
- It is enough to check for all generators.

example1:

Bell state

$$\mathcal{S}_1 = \langle XX, ZZ \rangle \longrightarrow (|00\rangle + |11\rangle)/\sqrt{2}$$

example2:

subspace spanned by

$$\mathcal{S}_2 = \langle ZZ \rangle \longrightarrow \{|00\rangle, |11\rangle\}$$

# of qubit *n*, # of stabilizer generators *k*,
the dimension of the stabilizer subspace → $d = 2^{n-k}$

“Heisenberg picture of quantum computation”

Clifford operator：Map a Pauli product to another Pauli product

→ a stabilizer state to another stabilizer state.

[Heisenberg] **[Schrödinger]**

$$
\begin{tikzcd}
\langle S_i \rangle \arrow[rr, leftrightarrow, "S_i|\Psi\rangle = |\Psi\rangle"'] \arrow[d, "U"'] & & |\Psi\rangle \arrow[d, "U"] \\
S_i' = U S_i U^{\dagger} \quad \langle S_i' \rangle \arrow[rr, leftrightarrow, "S_i'|\Psi'\rangle = |\Psi'\rangle"'] & & |\Psi'\rangle = U|\Psi\rangle
\end{tikzcd}
$$

“Heisenberg picture of quantum computation”

Clifford operator：Map a Pauli product to another Pauli product

→ a stabilizer state to another stabilizer state.

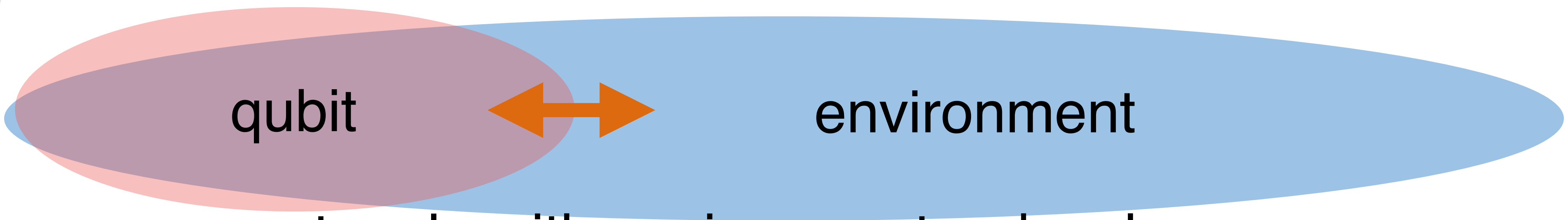
[Heisenberg] **[Schrödinger]**

$$
\begin{tikzcd}
\langle S_i \rangle \arrow[rr, leftrightarrow, "S_i|\Psi\rangle = |\Psi\rangle"'] \arrow[d, "U"'] & & |\Psi\rangle \arrow[d, "U"] \\
S_i' = U S_i U^\dagger \quad \langle S_i' \rangle \arrow[rr, leftrightarrow, "S_i'|\Psi'\rangle = |\Psi'\rangle"'] & & |\Psi'\rangle = U|\Psi\rangle
\end{tikzcd}
$$

$$\boxed{S_i \to S_i' = U S_i U^\dagger}$$

Application of stabilizer formalism: Quantum error correction codes

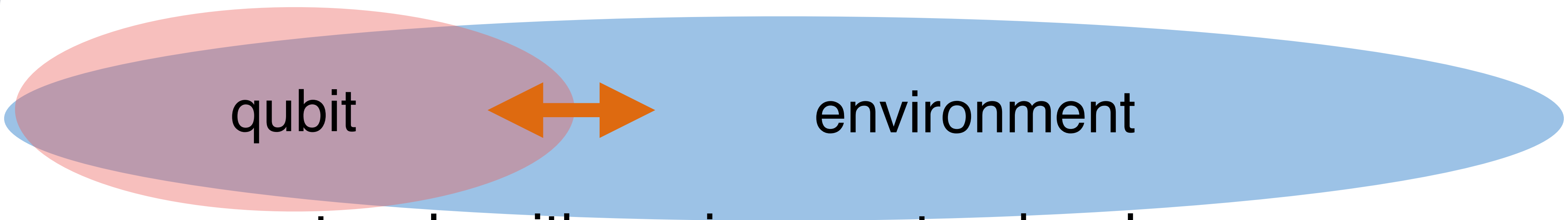
Quantum error correction codes



entangle with environment = decoherence

$$\alpha|0\rangle + \beta|1\rangle \to |\alpha|^2|0\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|$$

Quantum error correction codes



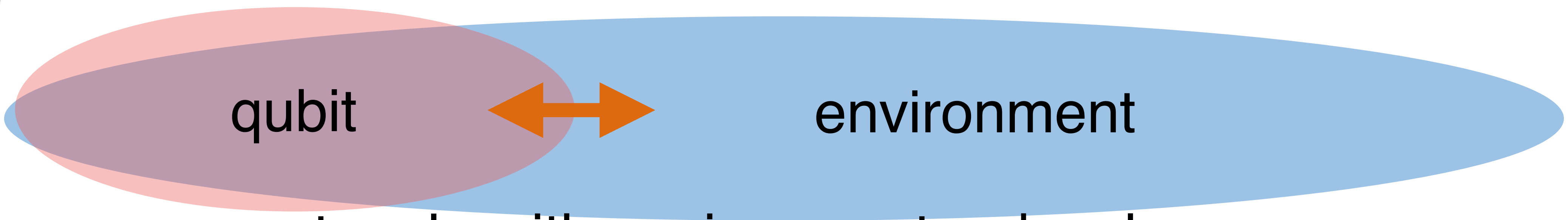
entangle with environment = decoherence

$$\alpha|0\rangle + \beta|1\rangle \to |\alpha|^2|0\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|$$

- Quantum state is parameterized by complex variables.
- no-cloning theorem→cannot copy it to protect

[Wootters-Zurek82]

Quantum error correction codes



entangle with environment = decoherence

$$\alpha|0\rangle + \beta|1\rangle \to |\alpha|^2|0\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|$$

- Quantum state is parameterized by complex variables.
- no-cloning theorem→cannot copy it to protect [Wootters-Zurek82]

Quantum error correction code [Shor95]

"fight entanglement with entanglement"

[Preskill97]

Peter Shor

http://www-math.mit.edu/~shor/

Quantum error correction code

Classical error correction： 0 → 000, 1 → 111

Quantum error correction：$\cancel{|\psi\rangle \to |\psi\rangle|\psi\rangle|\psi\rangle}$

$$\begin{cases} |0\rangle \to |000\rangle \\ |1\rangle \to |111\rangle \end{cases}$$

$$\alpha|0\rangle + \beta|1\rangle \to \alpha|000\rangle + \beta|111\rangle$$

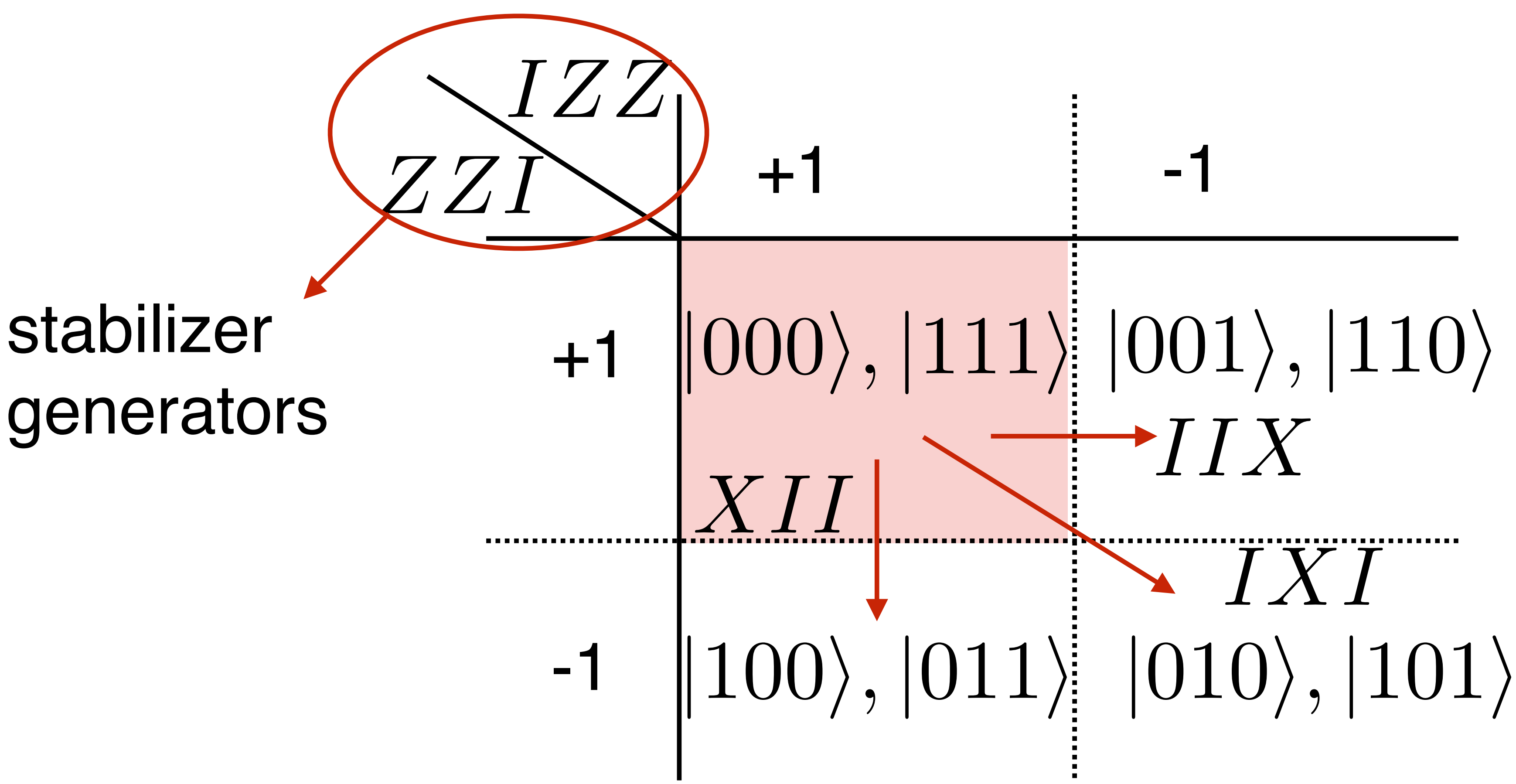
Quantum error correction code

Classical error correction：0 → 000, 1 → 111

Quantum error correction：$\cancel{|\psi\rangle \to |\psi\rangle|\psi\rangle|\psi\rangle}$

$$\begin{cases} |0\rangle \to |000\rangle \\ |1\rangle \to |111\rangle \end{cases}$$

$$\alpha|0\rangle + \beta|1\rangle \to \alpha|000\rangle + \beta|111\rangle$$



Quantum error correction code

Classical error correction： 0 → 000, 1 → 111

Quantum error correction： $\cancel{|\psi\rangle \rightarrow |\psi\rangle|\psi\rangle|\psi\rangle}$

$$\begin{cases} |0\rangle \rightarrow |000\rangle \\ |1\rangle \rightarrow |111\rangle \end{cases}$$

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|000\rangle + \beta|111\rangle$$

ZZI \ IZZ	+1	-1
+1	$\|000\rangle, \|111\rangle$	$\|001\rangle, \|110\rangle$
-1	$\|100\rangle, \|011\rangle$	$\|010\rangle, \|101\rangle$

stabilizer generators

IIX

XII

IXI

anti-commute with stabilizer generators
→map the state to the orthogonal space

but still superposition is preserved!

Quantum error correction code

Classical error correction : 0 → 000, 1 → 111

Quantum error correction : $\cancel{|\psi\rangle \to |\psi\rangle|\psi\rangle|\psi\rangle}$

$$\begin{cases} |0\rangle \to |000\rangle \\ |1\rangle \to |111\rangle \end{cases}$$

$$\alpha|0\rangle + \beta|1\rangle \to \alpha|000\rangle + \beta|111\rangle$$

$ZZI \backslash IZZ$	+1	-1
+1	$\|000\rangle, \|111\rangle$	$\|001\rangle, \|110\rangle$
-1	$\|100\rangle, \|011\rangle$	$\|010\rangle, \|101\rangle$

XII IIX IXI

stabilizer generators

anti-commute with stabilizer generators →map the state to the orthogonal space

but still superposition is preserved!

Protect quantum information from a single bit-flip error.

Stabilizer codes and logical operator

A stabilizer code is a quantum code defined as a stabilizer subspace:

$$|\Psi\rangle = S_i|\Psi\rangle \quad \text{for all } S_i \in \mathcal{S}_n$$

dim = 2^(# of qubit - # of generators)

Stabilizer codes and logical operator

A stabilizer code is a quantum code defined as a stabilizer subspace:

$$|\Psi\rangle = S_i|\Psi\rangle \quad \text{for all } S_i \in \mathcal{S}_n$$

dim = 2^(# of qubit - # of generators)

How can we specify the encoded (logical) degree of freedom?

Stabilizer codes and logical operator

A stabilizer code is a quantum code defined as a stabilizer subspace:

$$|\Psi\rangle = S_i|\Psi\rangle \quad \text{for all } S_i \in \mathcal{S}_n$$

dim = 2^(# of qubit - # of generators)

How can we specify the encoded (logical) degree of freedom?

Logical operators: commute with & independent of the stabilizer group

ex) $\mathcal{S} = \langle ZZI, IZZ \rangle, \;\; L_X = XXX, L_Z = IIZ$

$\rightarrow \{|000\rangle, |111\rangle\} \quad IIZ|111\rangle = -|111\rangle, \;\; XXX|000\rangle = |111\rangle$

logical operators act nontrivially inside the code space

5-qubit code

stabilizer operators

$S_1 = ZXXZI$
$S_2 = IZXXZ$
$S_3 = ZIZXX$
$S_4 = XZIZX$

For all stabilizers,

$$S_i|\Psi_L\rangle = |\Psi_L\rangle$$

$2^5 / 2^4 = 2$ dimensional subspace

5-qubit code

stabilizer operators
$S_1 = ZXXZI$
$S_2 = IZXXZ$
$S_3 = ZIZXX$
$S_4 = XZIZX$

For all stabilizers,

$$S_i|\Psi_L\rangle = |\Psi_L\rangle$$

$2^5 / 2^4 = 2$ dimensional subspace

→Logical Pauli operators

$$X_L = X^{\otimes 5}, Z_L = Z^{\otimes 5}$$

act on the code space nontrivially.

5-qubit code

stabilizer operators

$S_1 = ZXXZI$
$S_2 = IZXXZ$
$S_3 = ZIZXX$
$S_4 = XZIZX$

For all stabilizers,

$$S_i|\Psi_L\rangle = |\Psi_L\rangle$$

$2^5 / 2^4 = 2$ dimensional subspace

→Logical Pauli operators

$$X_L = X^{\otimes 5}, Z_L = Z^{\otimes 5}$$

act on the code space nontrivially.

Anti-commute with any single Pauli error, X,Y Z.

→**an arbitrary single-qubit error is corrected.**

5-qubit code

stabilizer operators

$$S_1 = ZXXZI$$
$$S_2 = IZXXZ$$
$$S_3 = ZIZXX$$
$$S_4 = XZIZX$$

2^5=32 dimensions

2^4=16 orthogonal subspace

{X,Y,Z}×5 qubits = 15

15+1=16

For all stabilizers,

$$S_i|\Psi_L\rangle = |\Psi_L\rangle$$

$2^5 / 2^4$ = 2 dimensional subspace

→Logical Pauli operators

$$X_L = X^{\otimes 5}, Z_L = Z^{\otimes 5}$$

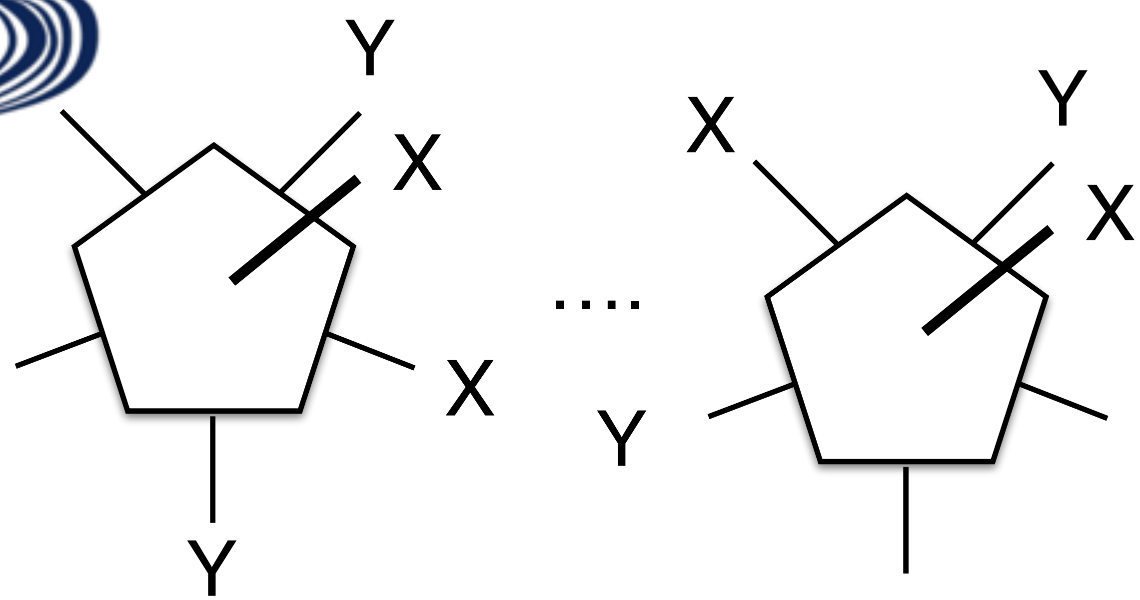
act on the code space nontrivially.

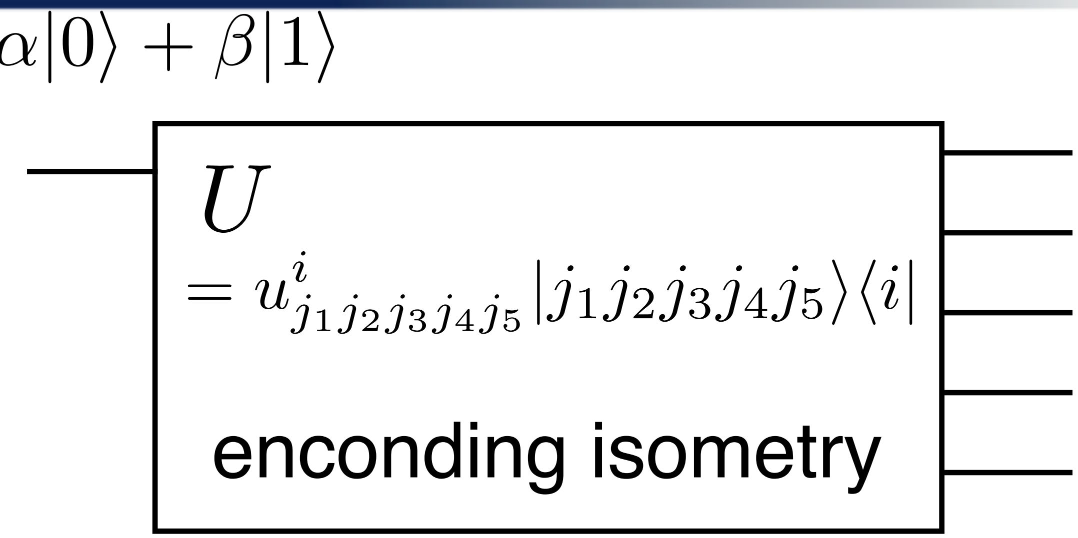
Anti-commute with any single Pauli error, X,Y Z.

→**an arbitrary single-qubit error is corrected.**

# of errors = # of orthogonal subspaces!

A toy model for AdS/CFT





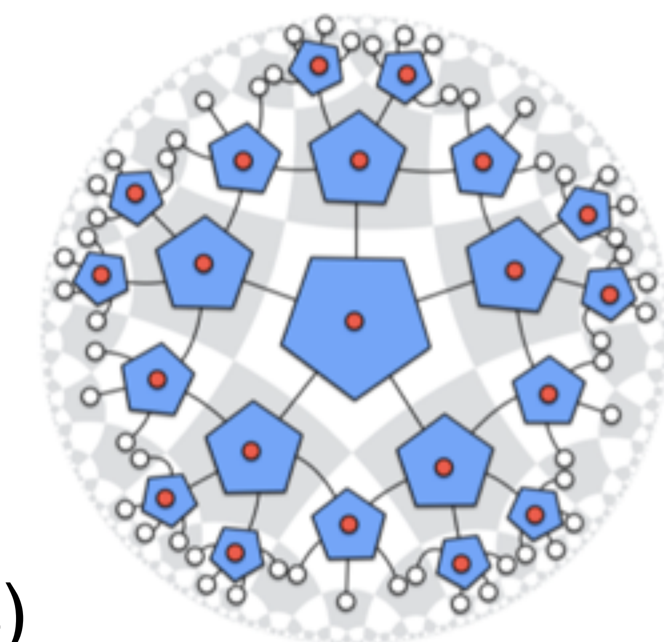
2-dim subspace of 2^5-dim space

stabilizers act trivially on the code space
→equivalent class of logical operators

- Ryu-Takayanagi formula
- AdS-Rindler/ causal wedge reconstruction

  (boundary reconstruction of bulk operators)

physical qubits logical qubits



(b) Holographic pentagon code

[HaPPY, JHEP '15]

Application of stabilizer formalism: measurement-based QC

Graph (cluster) state

◆ Definition of a graph state

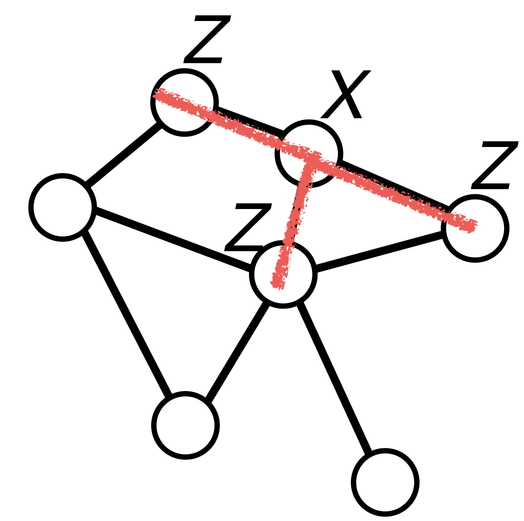
A stabilizer generator is defined for each vetex

$$K_i = X_i \prod_{j \sim i} Z_j$$

$$K_i |G\rangle = |G\rangle \text{ for all } i \in V$$

graph G=(V,E)

V: vertices,   E:edges



Graph (cluster) state

- Definition of a graph state

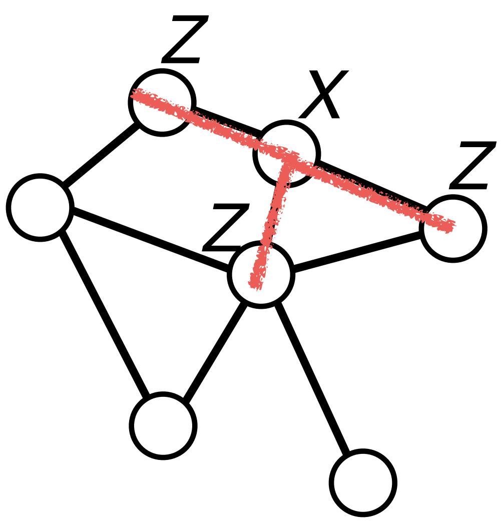
A stabilizer generator is defined for each vetex

$$K_i = X_i \prod_{j \sim i} Z_j$$

$$K_i|G\rangle = |G\rangle \text{ for all } i \in V$$

graph G=(V,E)

V: vertices,   E:edges



CZ gate

$$|G\rangle = \prod_{e \in E} \Lambda_e(Z)|+\rangle^{\otimes |V|}$$

Graph (cluster) state

- Definition of a graph state

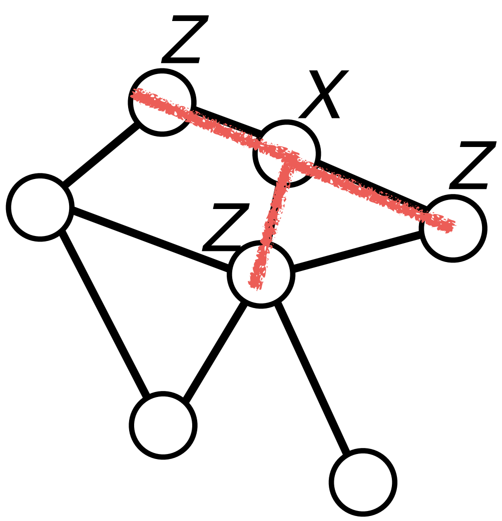
A stabilizer generator is defined for each vetex

$$K_i = X_i \prod_{j \sim i} Z_j$$

$$K_i|G\rangle = |G\rangle \text{ for all } i \in V$$

graph G=(V,E)

V: vertices, E:edges



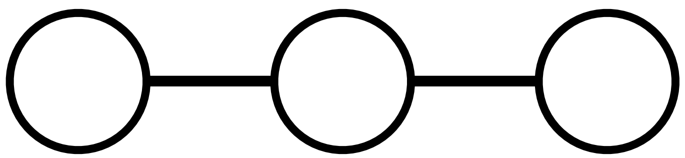
CZ gate

$$|G\rangle = \prod_{e \in E} \Lambda_e(Z) |+\rangle^{\otimes |V|}$$

$$K_i = \left[\prod_{e \in E} \Lambda(Z)\right] X_i \left[\prod_{e \in E} \Lambda(Z)\right]$$

1D graph (cluster) state

- 3-qubit 1D graph state

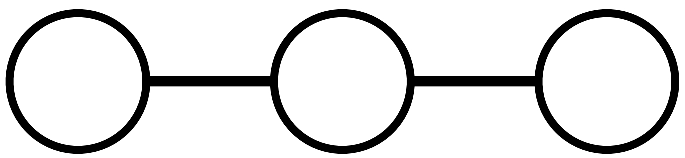


$$\begin{array}{ccccc} X & — & Z & & \\ Z & — & X & — & Z \\ & & Z & — & X \end{array}$$

$$\frac{1}{\sqrt{2}}(|+\rangle|0\rangle|+\rangle + |-\rangle|1\rangle|-\rangle)$$

1D graph (cluster) state

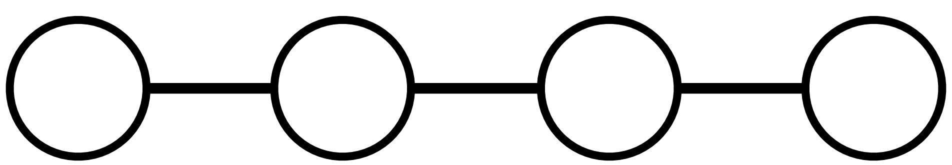
- 3-qubit 1D graph state



$$\begin{array}{ccccc} X & \text{---} & Z & & \\ Z & \text{---} & X & \text{---} & Z \\ & & Z & \text{---} & X \end{array}$$

$$\frac{1}{\sqrt{2}}(|+\rangle|0\rangle|+\rangle + |-\rangle|1\rangle|-\rangle)$$

- 4-qubit 1D graph state



$$\begin{array}{ccccccc} X & \text{---} & Z & & & & \\ Z & \text{---} & X & \text{---} & Z & & \\ & & Z & \text{---} & X & \text{---} & Z \\ & & & & Z & \text{---} & X \end{array}$$

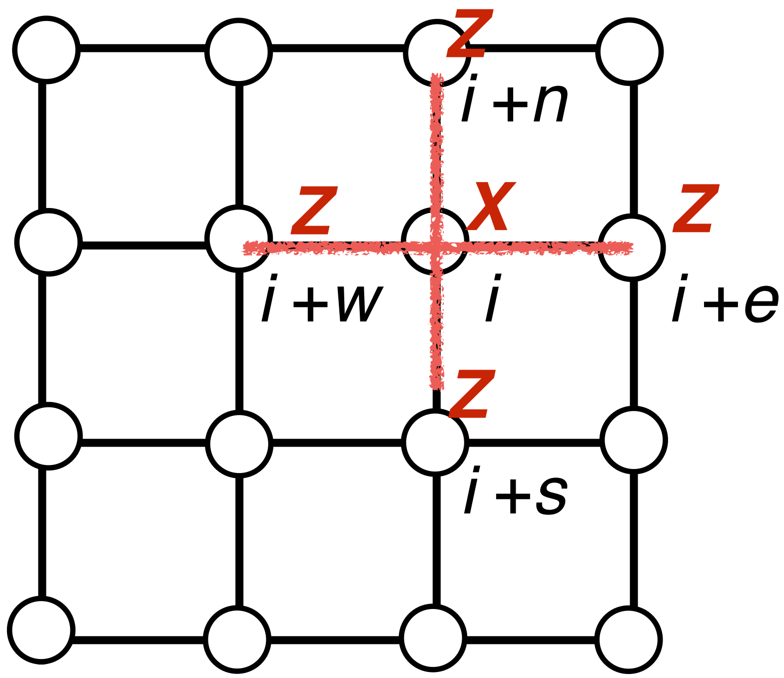
$$\frac{1}{2}(|+\rangle|0\rangle|0\rangle|+\rangle + |+\rangle|0\rangle|1\rangle|-\rangle$$
$$+ |-\rangle|1\rangle|0\rangle|+\rangle - |-\rangle|1\rangle|1\rangle|-\rangle)$$

MBQC

measurement-based quantum computation

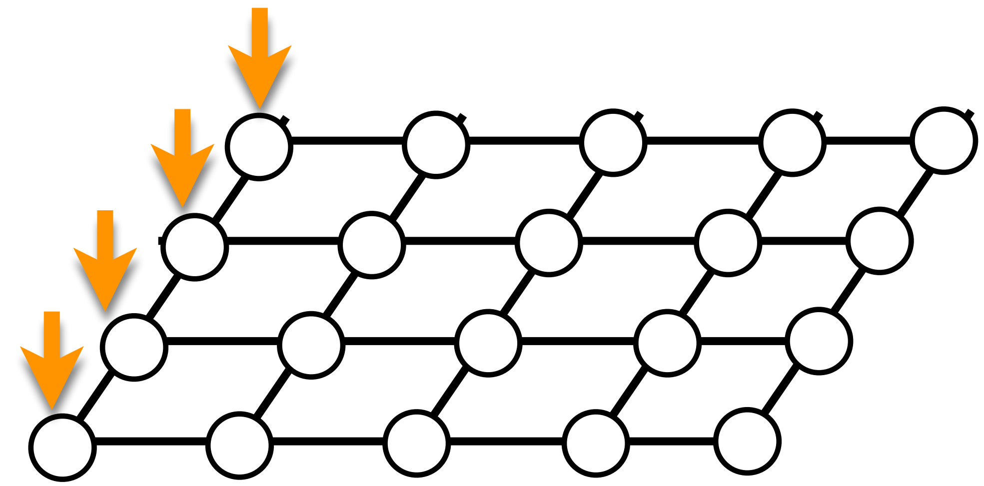
Raussendorf-Briegel PRL **86** 910 (2001); Raussendorf-Browne-Briegel PRA **68** 022312 (2003).

- 2D cluster state



$$K_i = X_i Z_{i+n} Z_{i+e} Z_{i+s} Z_{i+w}$$

projective measurement



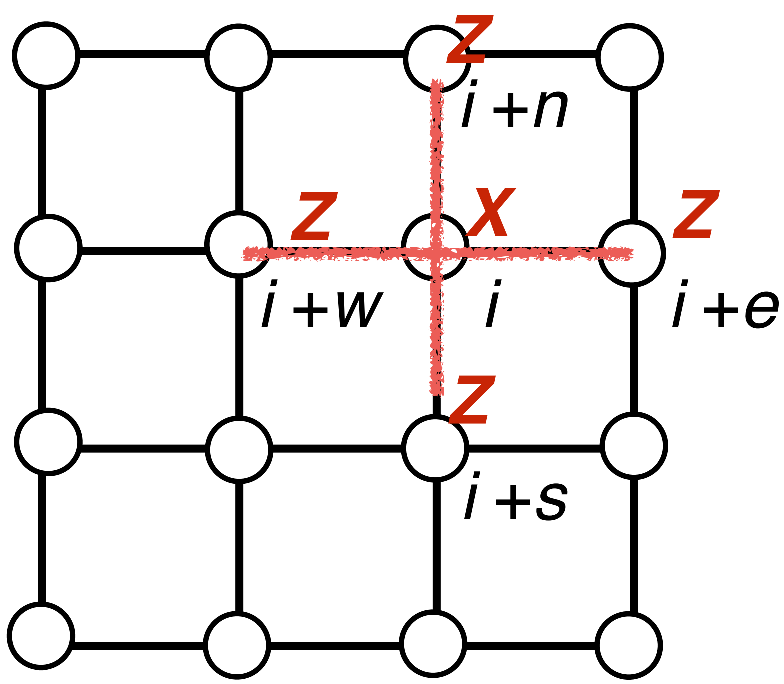
2D resource state

MBQC

measurement-based quantum computation

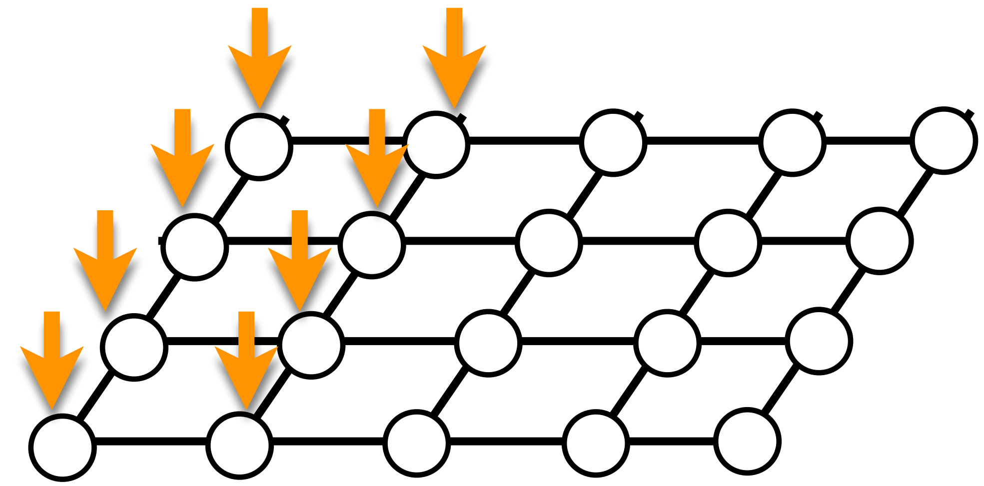
Raussendorf-Briegel PRL **86** 910 (2001); Raussendorf-Browne-Briegel PRA **68** 022312 (2003).

- 2D cluster state



$$K_i = X_i Z_{i+n} Z_{i+e} Z_{i+s} Z_{i+w}$$

projective measurement



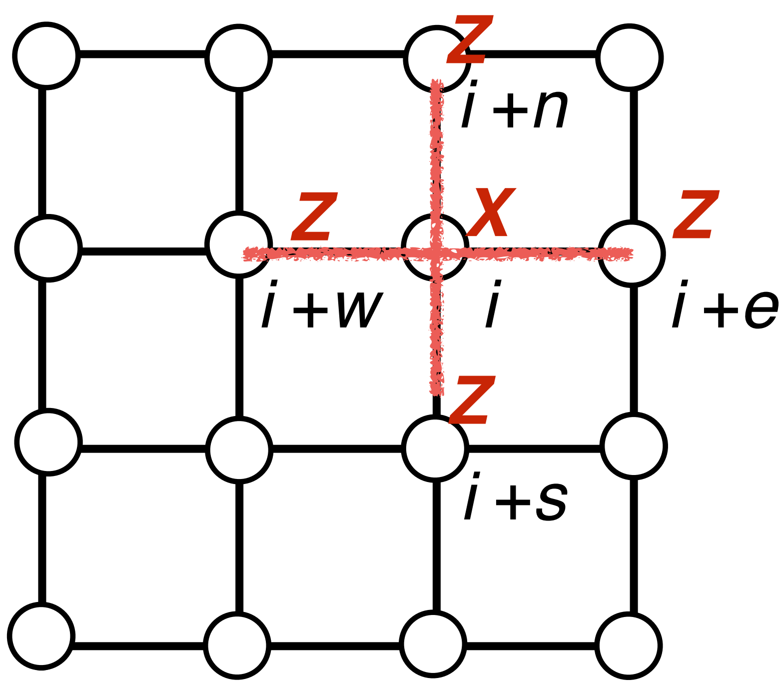
2D resource state

MBQC

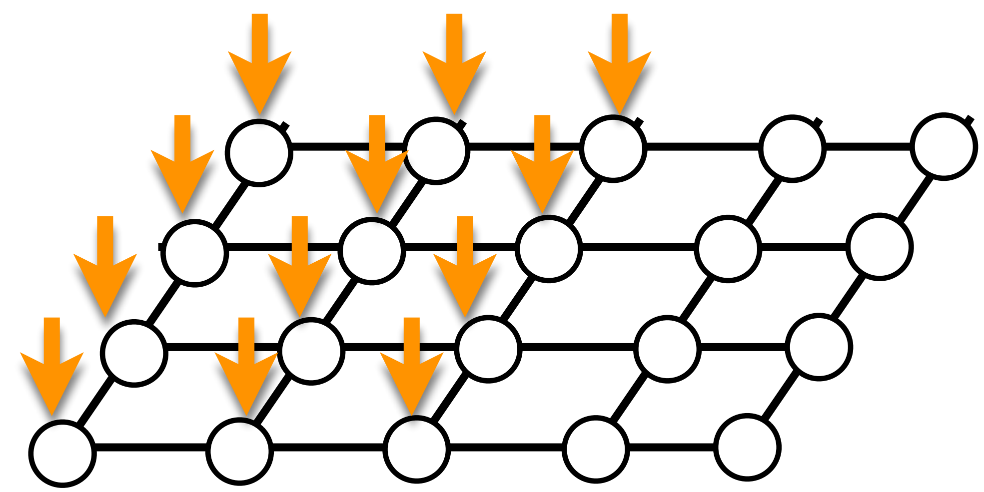
measurement-based quantum computation

Raussendorf-Briegel PRL **86** 910 (2001); Raussendorf-Browne-Briegel PRA **68** 022312 (2003).

- 2D cluster state



$$K_i = X_i Z_{i+n} Z_{i+e} Z_{i+s} Z_{i+w}$$

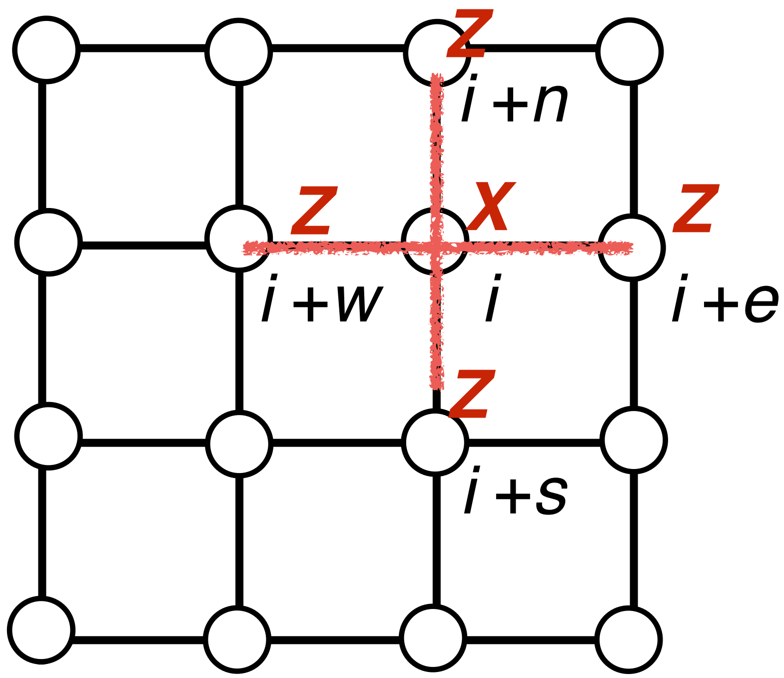


MBQC

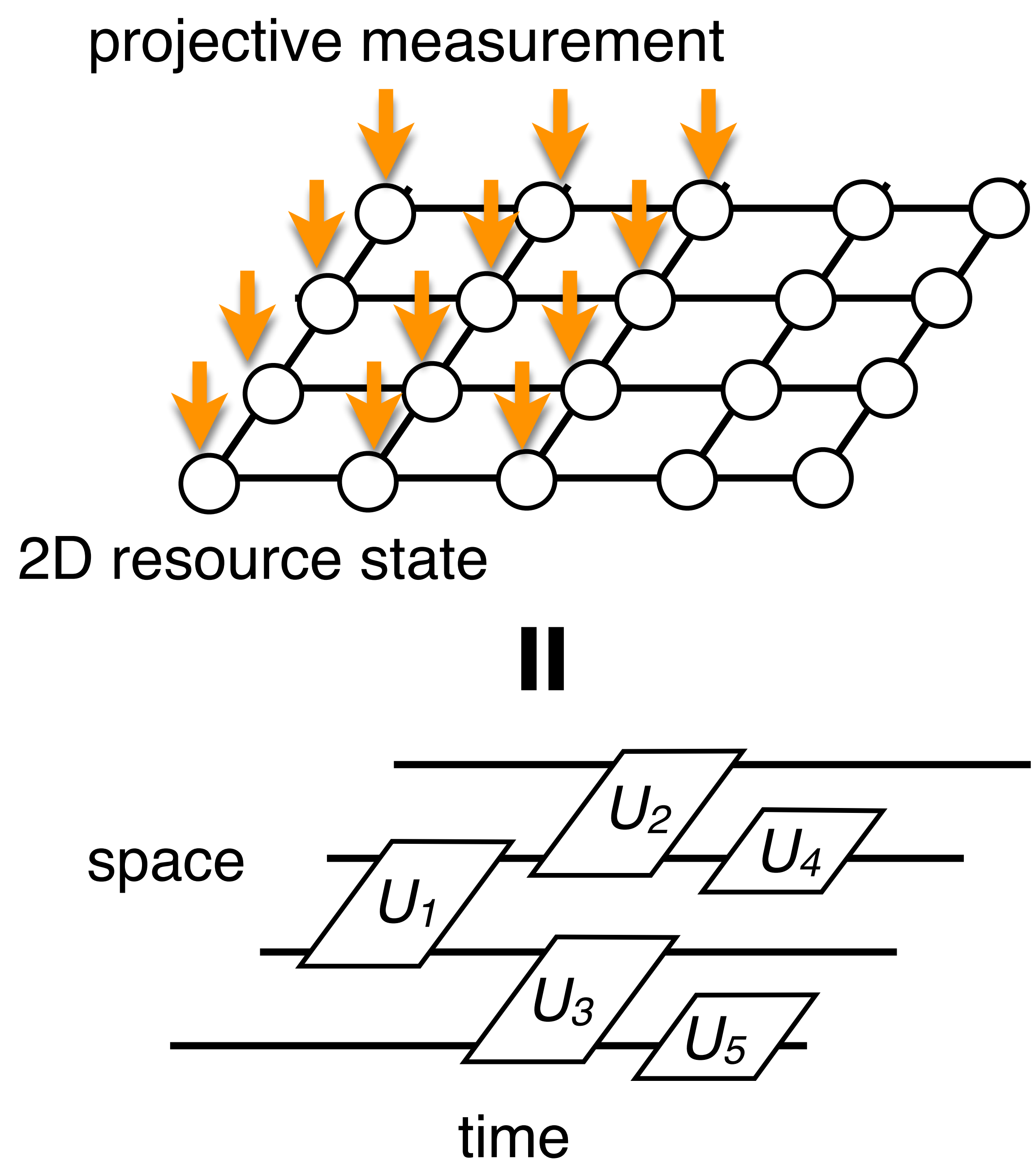
measurement-based quantum computation

Raussendorf-Briegel PRL **86** 910 (2001); Raussendorf-Browne-Briegel PRA **68** 022312 (2003).

- 2D cluster state



$$K_i = X_i Z_{i+n} Z_{i+e} Z_{i+s} Z_{i+w}$$

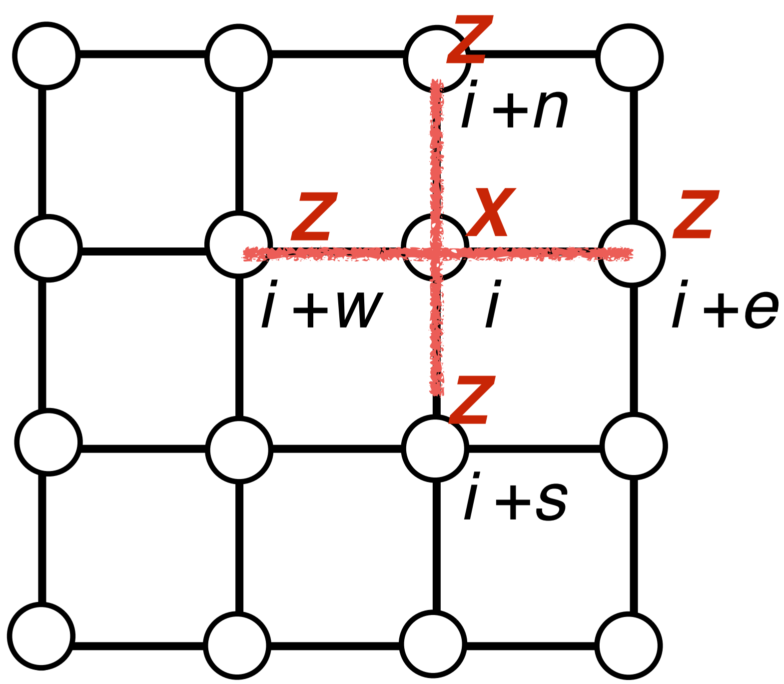


MBQC

measurement-based quantum computation

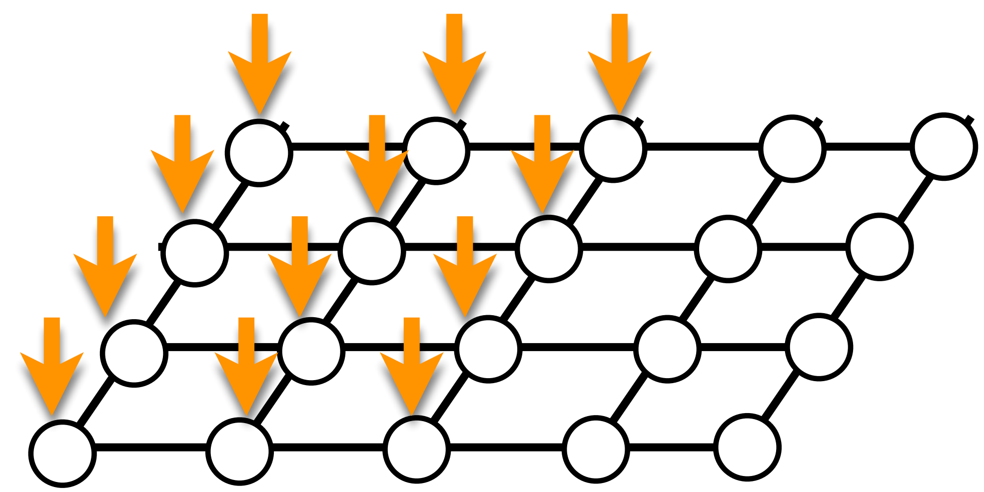
Raussendorf-Briegel PRL **86** 910 (2001); Raussendorf-Browne-Briegel PRA **68** 022312 (2003).

◆ 2D cluster state



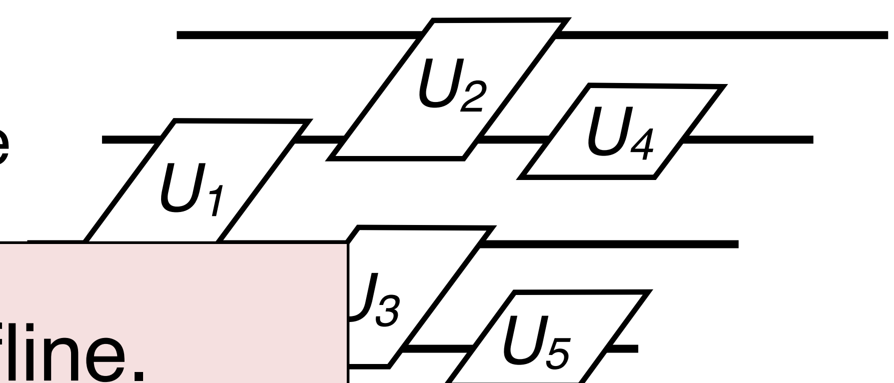
$$K_i = X_i Z_{i+n} Z_{i+e} Z_{i+s} Z_{i+w}$$

projective measurement



2D resource state

=



- Entangling operations are required only offline.
- Provide a connection between many-body physics.

Summary

How quantum computer works

→H, π/8gate, CNOT, Solovay-Kitaev algo.

How quantum algorithm works

→estimation of eigenvalues of unitary operators

How quantum states are efficiently described

→ not by the state but the operators that stabilize the state