Quantum Computing with Noninteracting Particles

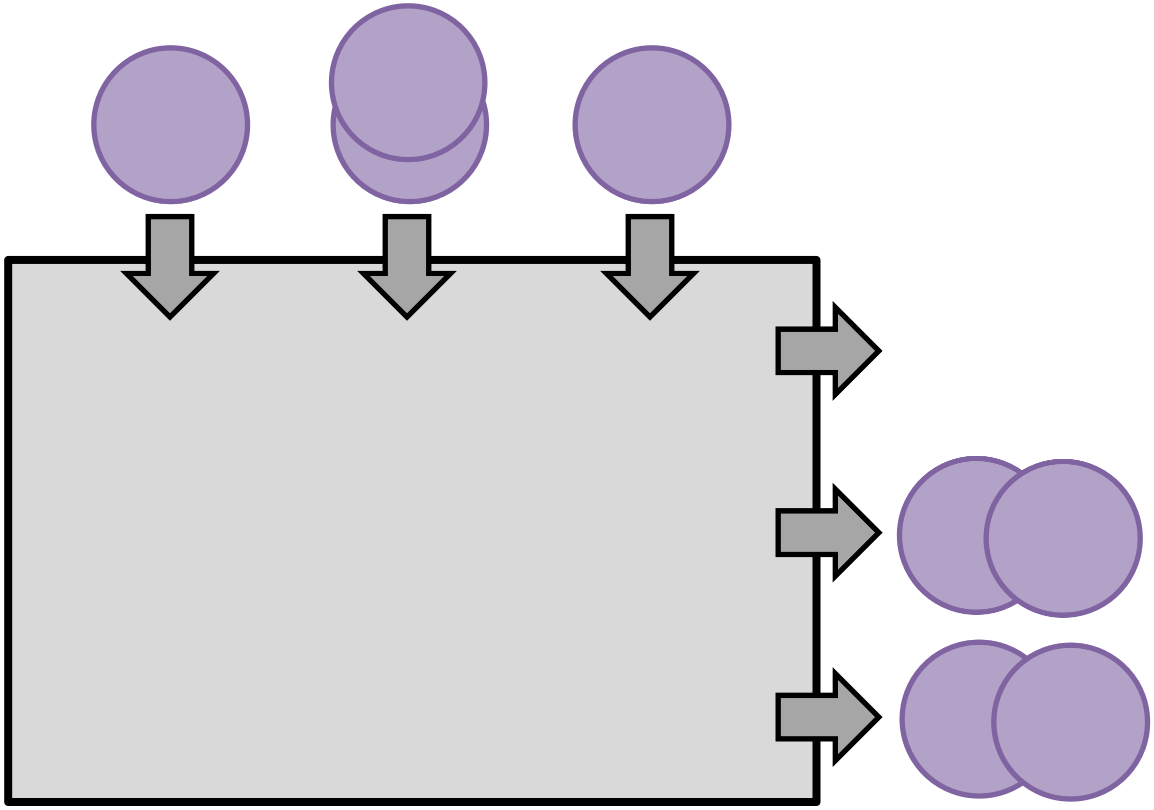
Alex Arkhipov

Noninteracting Particle Model

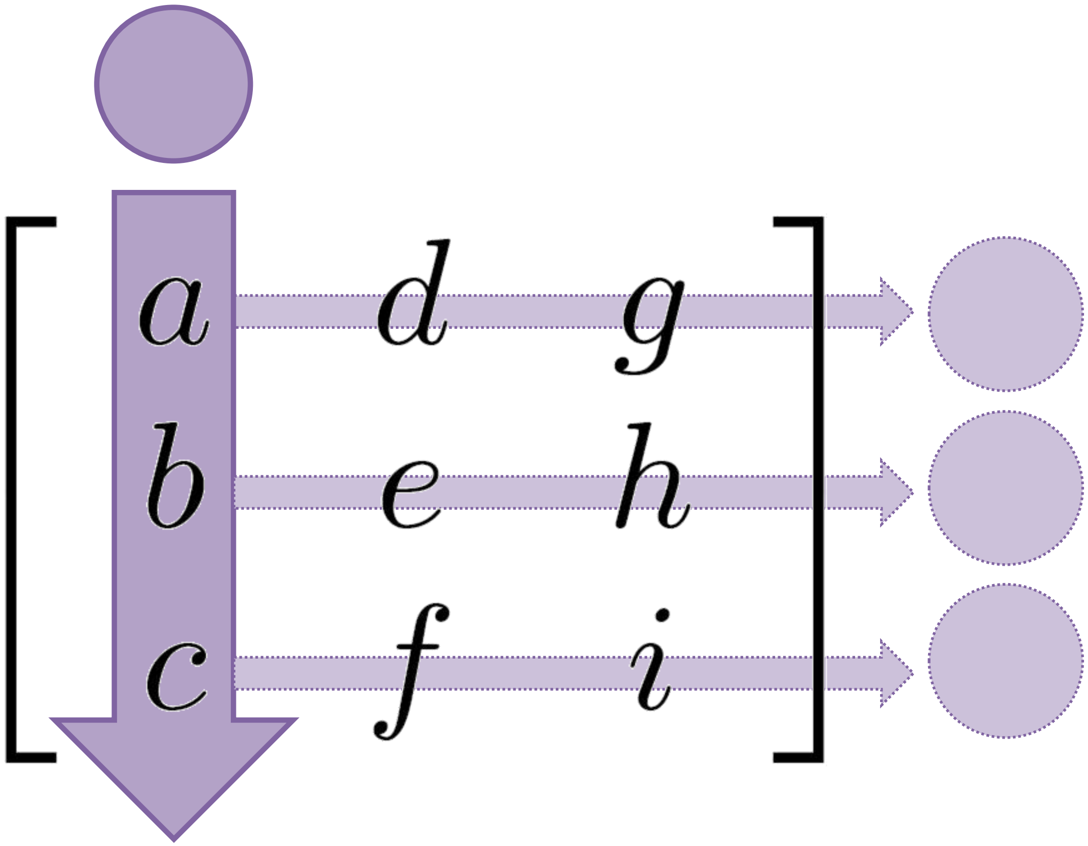
- Weak model of QC
  - Probably not universal
  - Restricted kind of entanglement
  - Not qubit-based
- Why do we care?
  - Gains with less quantum
  - Easier to build
  - Mathematically pretty

Classical Analogue

Balls and Slots

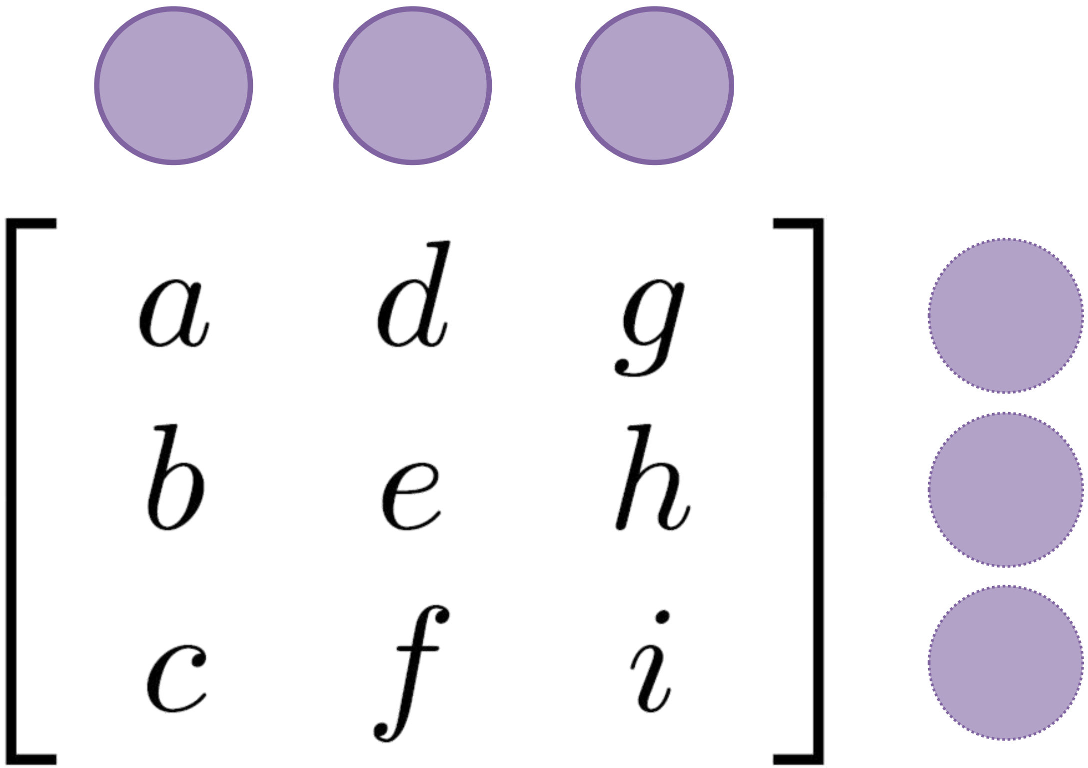


Transition Matrix



$$a, b, c \geq 0$$
$$a + b + c = 1$$

A Transition Probability

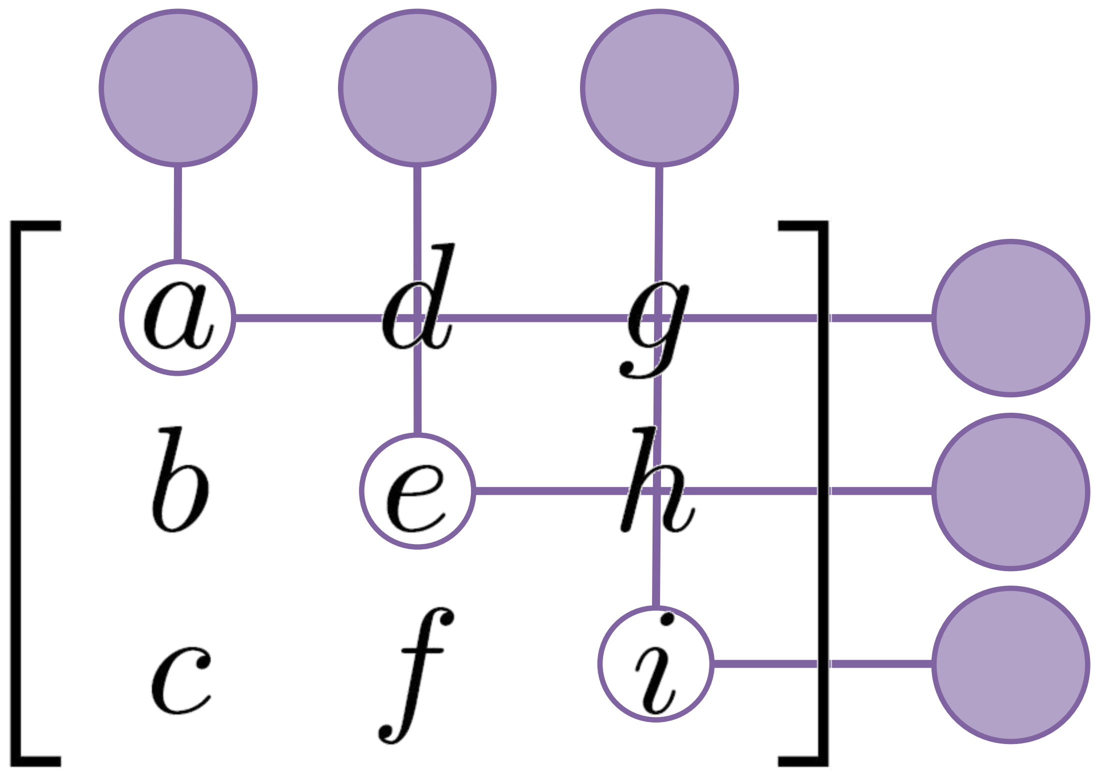


$$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

?

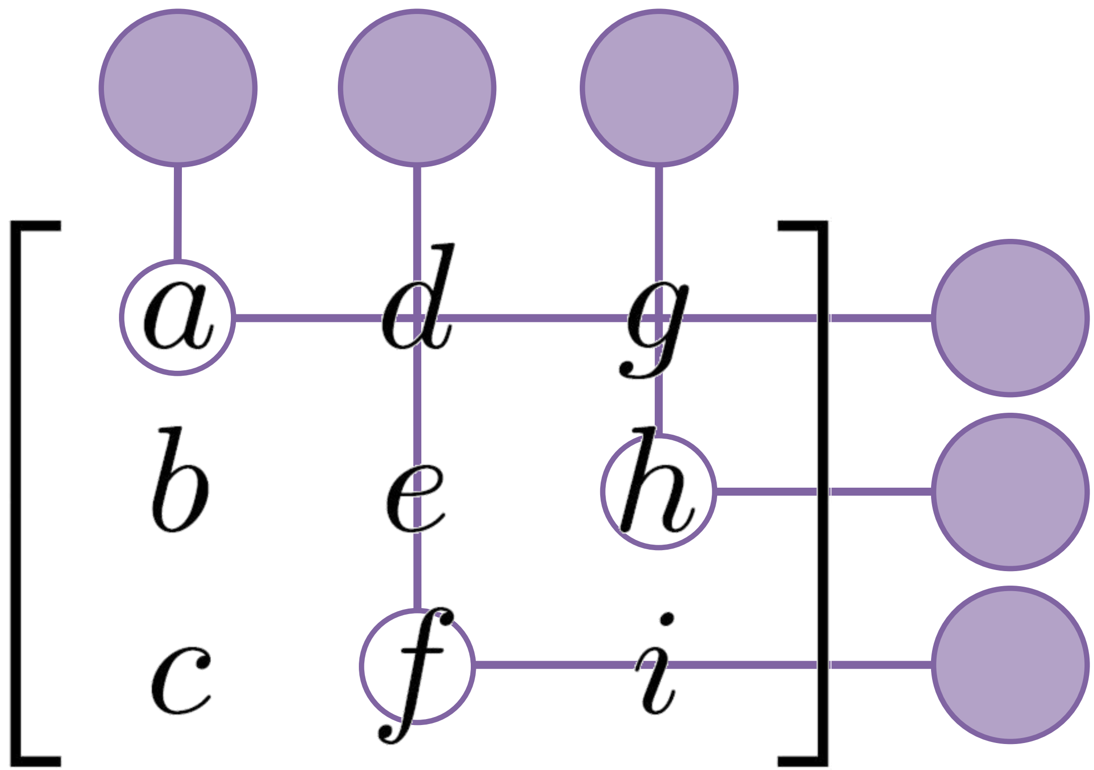
$$\Pr[\text{one per slot}] =$$

A Transition Probability



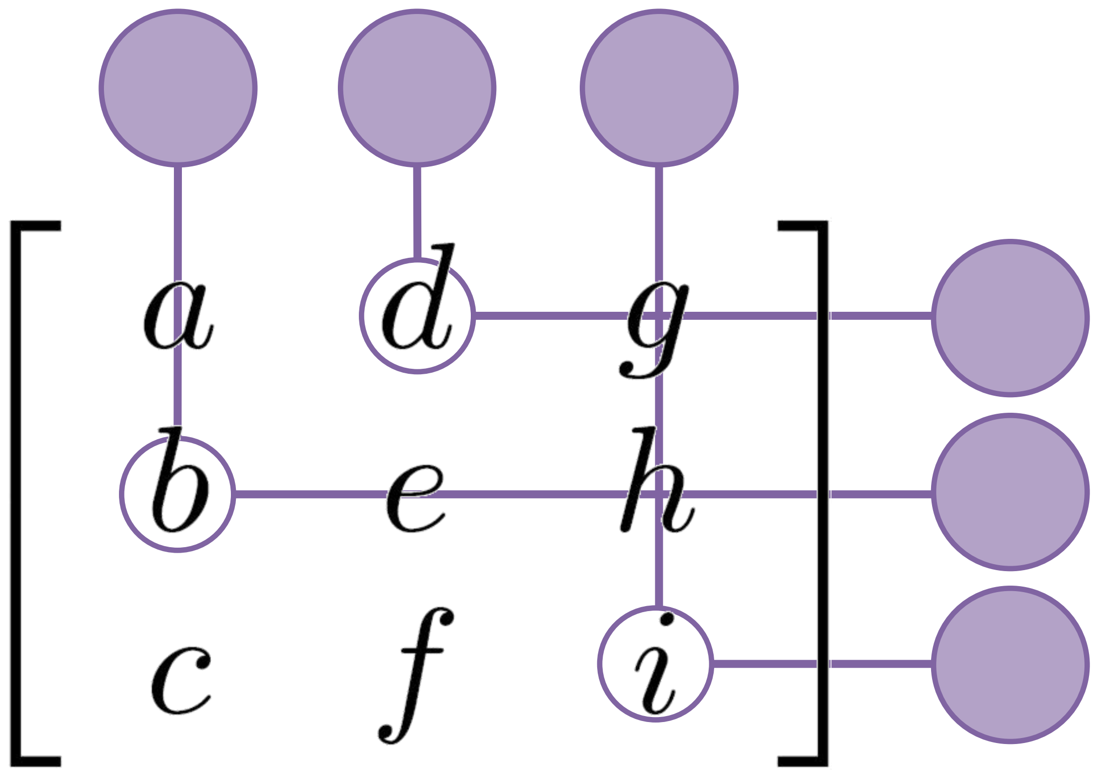
$\Pr[\text{one per slot}] = aei +$

A Transition Probability



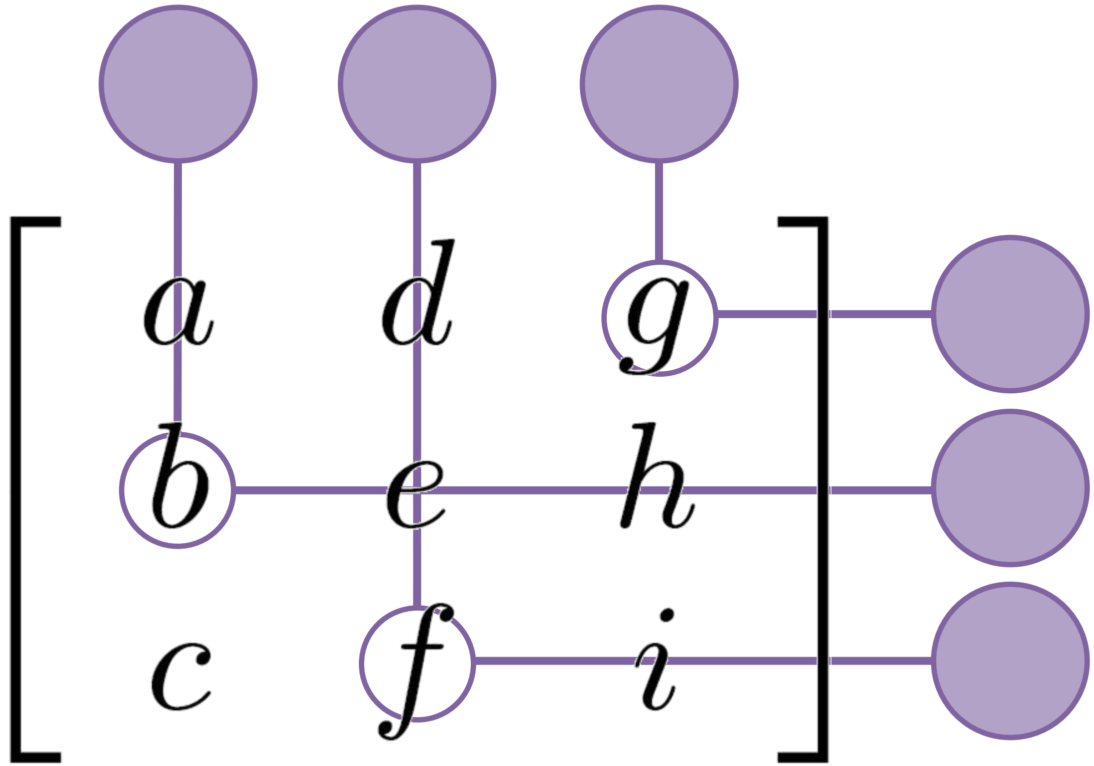
$$\Pr\left[\text{one per slot}\right] = aei + afh$$

A Transition Probability



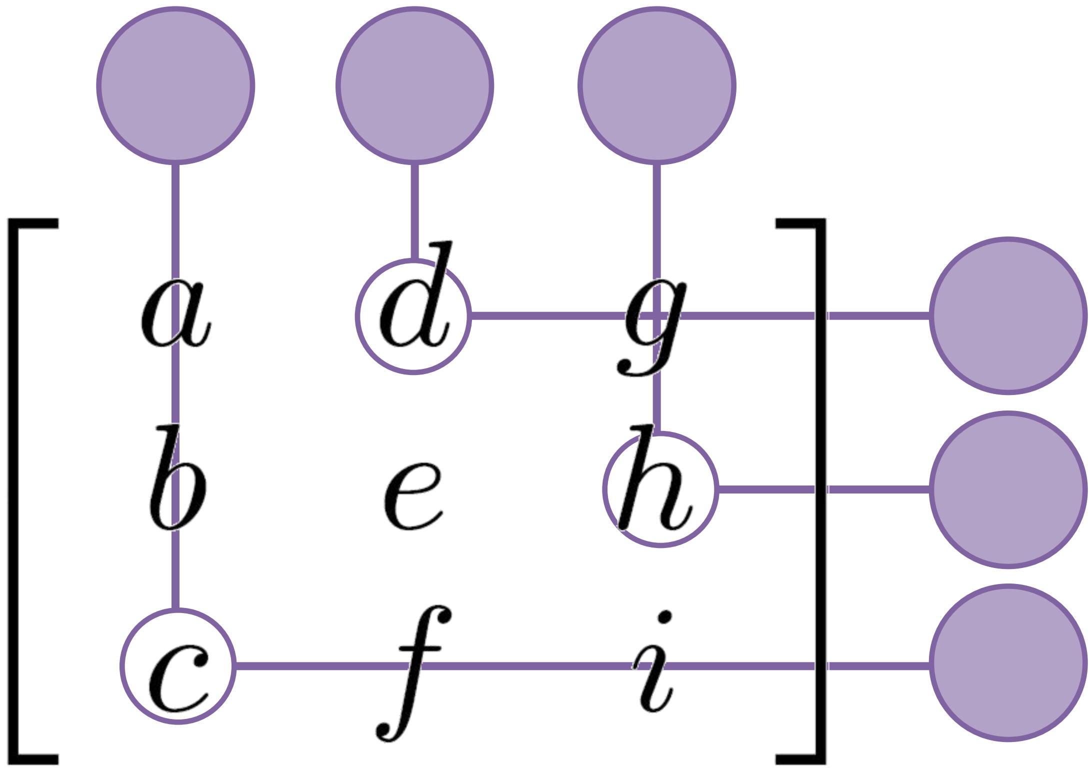
$$\Pr[\text{one per slot}] = aei + afh + bdi$$

A Transition Probability



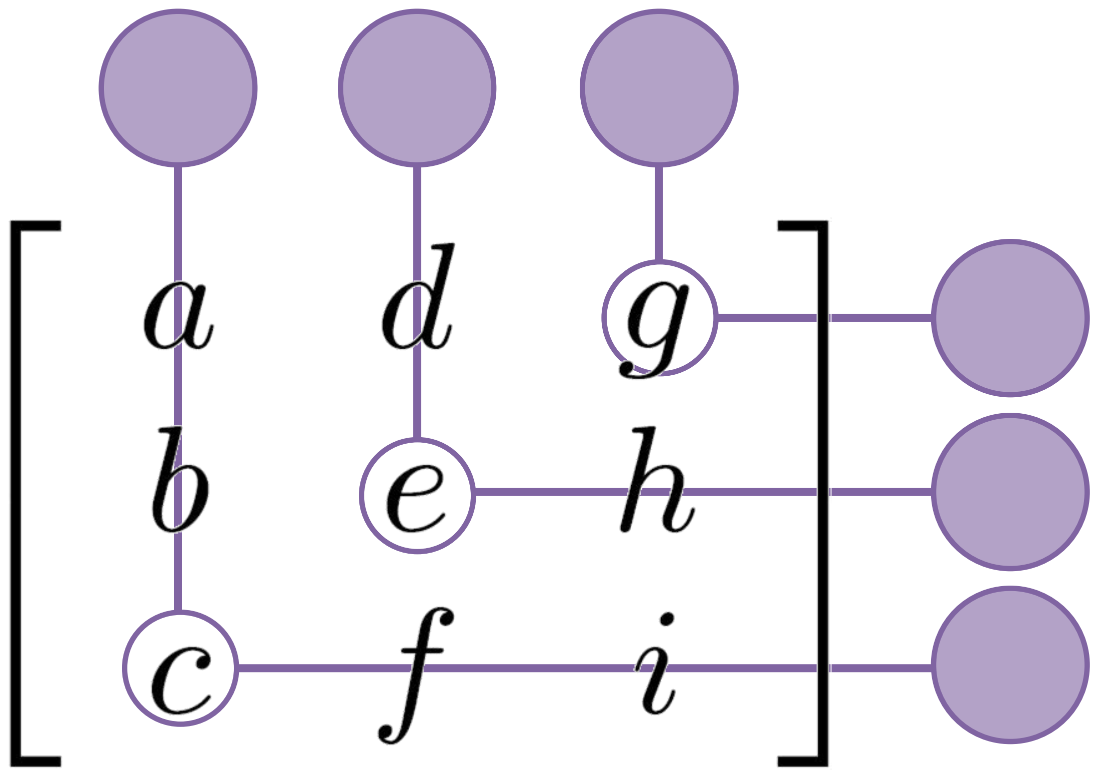
$$\Pr\left[\text{one per slot}\right] = aei + afh + bdi + bfg$$

A Transition Probability



$$\Pr\left[\text{one per slot}\right] = aei + afh + bdi + bfg + cdh$$

A Transition Probability



$$\Pr\left[\text{one per slot}\right] = aei + afh + bdi + bfg + cdh + ceg$$

$$= \operatorname{perm}(M)$$

Probabilities for Classical Analogue

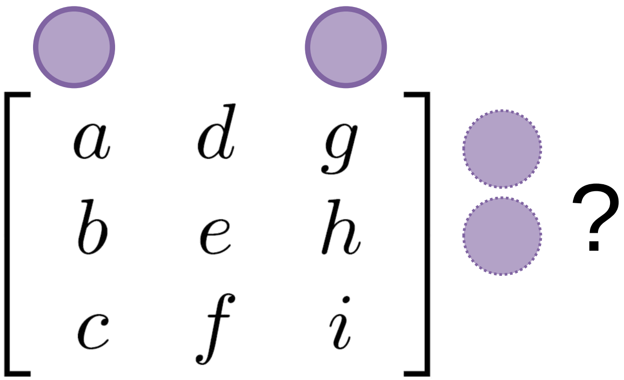
$$\Pr\left[\text{one per slot} \to \text{one per slot}\right] = \sum_{\sigma \in S_n} \prod_{i=1}^{n} M_{\sigma(i),i}$$

$$\operatorname{perm}(M) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} M_{\sigma(i),i}$$

$$\det(M) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} M_{\sigma(i),i}$$

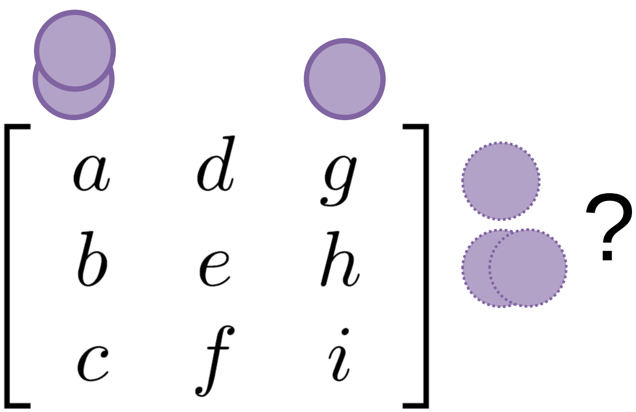
Probabilities for Classical Analogue

- What about other transitions?



$$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} \; ?$$

$$\mathrm{perm}\begin{bmatrix} a & g \\ b & h \end{bmatrix}$$



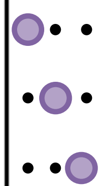
$$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} \; ?$$

$$\frac{1}{2}\mathrm{perm}\begin{bmatrix} a & a & g \\ b & b & h \\ b & b & h \end{bmatrix}$$

Configuration Transitions

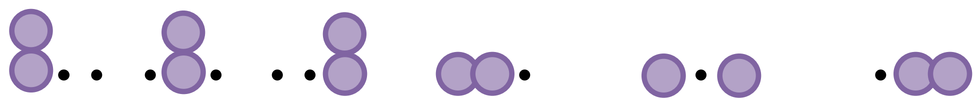
Transition matrix for one ball

$$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

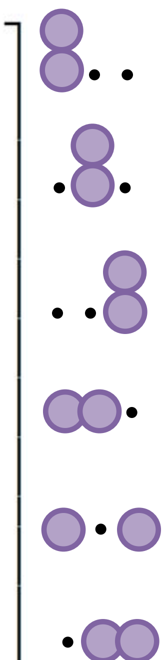


m=3, n=1

Transition matrix for two-ball configurations



$$\begin{bmatrix} a^2 & d^2 & g^2 & ad & ag & dg \\ b^2 & e^2 & h^2 & be & bh & eh \\ c^2 & f^2 & i^2 & cf & ci & fi \\ 2ab & 2de & 2gh & ae+bd & ah+bg & dh+eg \\ 2ac & 2df & 2gi & af+cd & ai+cg & ei+fh \\ 2bc & 2ef & 2hi & bf+ce & bi+ch & di+fg \end{bmatrix}$$



m=3, n=2

Classical Model Summary

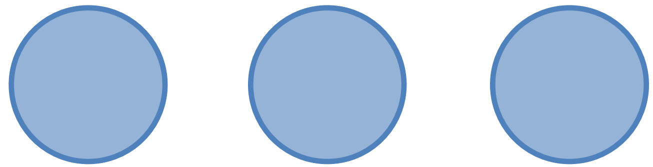
- n identical balls
- m slots
- Choose start **configuration**
- Choose stochastic **transition matrix** M
- Move each ball as per M
- Look at resulting configuration

Quantum Model

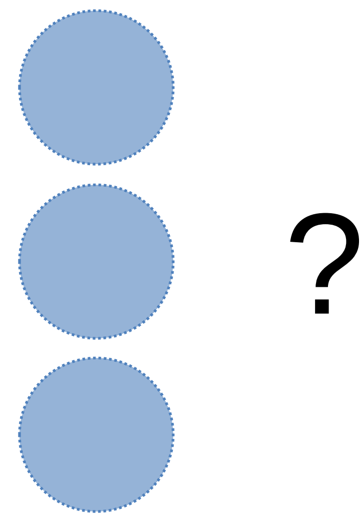
Quantum Particles

- Two types of particle: Bosons and Fermions

Identical Bosons

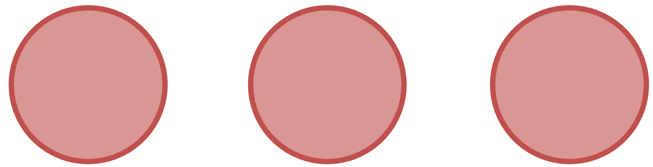


$$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

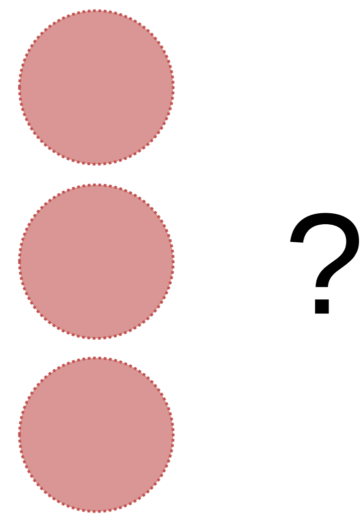


$$\begin{aligned} \text{Am}\left[\text{one per slot}\right] &= aei + afh + bdi + bfg + cdh + ceg \\ &= \text{perm}\left(M\right) \\ \Pr\left[\text{one per slot}\right] &= \left|\text{perm}\left(M\right)\right|^2 \end{aligned}$$

Identical Fermions



$$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$



$$\text{Am}\left[\text{one per slot}\right] = aei - afh - bdi + bfg + cdh - ceg$$

$$= \det\left(M\right)$$

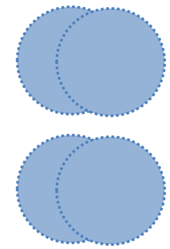
$$\Pr\left[\text{one per slot}\right] = \left|\det\left(M\right)\right|^2$$

Algebraic Formalism

- Modes are single-particle basis states
  - Variables $x_1, \ldots, x_m$
- Configurations are multi-particle basis states
  - Monomials $x_1^{a_1} x_2^{a_2} \cdots x_m^{a_m} / \sqrt{a_1! \cdots a_m!}$
- Identical bosons commute
  - $x_i x_j = x_j x_i$
- Identical fermions anticommute
  - $x_i x_j = -x_j x_i$
  - $x_i^2 = 0$

Example: Hadamarding Bosons

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



$$p = \frac{1}{2}$$

$$p = \frac{1}{2}$$

$$xy \Rightarrow \left(\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}}\right)\left(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}}\right)$$

$$= \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}x^2\right] - \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}y^2\right]$$

Hong-Ou-Mandel dip

Example: Hadamarding Fermions

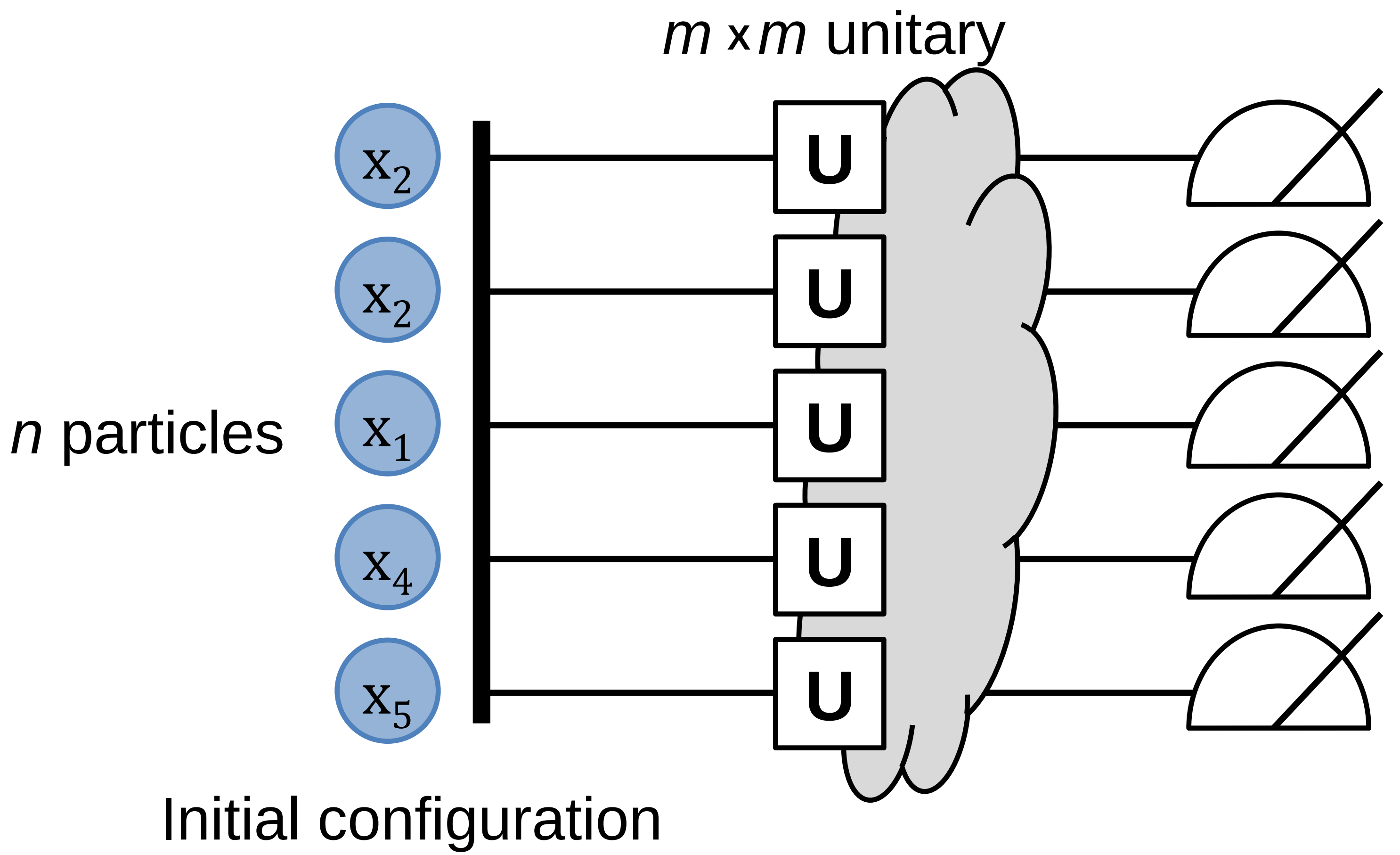
$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$xy \Rightarrow \left(\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}}\right)\left(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}}\right)$$

$$= \frac{1}{2}x^2 + \frac{1}{2}xy - \frac{1}{2}yx + \frac{1}{2}y^2$$

$$= xy$$

Definition of Model



Complexity

Complexity Comparison

Particle:			
Function:			
Matrix:			
Compute probability:			
Sample:			

Adaptive $\rightarrow$ BQP
[KLM '01]

Bosons Have the Hard Job

- Fermions: Easy
  - Det is in P
  - Doable in P [Valiant '01]
- Classical particles: Easy
  - Perm is #P-complete!
  - Perm approximable for ≥0 matrices [JSV '01]
- Bosons: Hard
  - With adaptive measurements, get BQP [KLM '01]
  - Not classically doable, even approximately [AA '10 in prep]

Bosons are Hard: Proof

- Classically simulate identical bosons

Approx counting

- Using NP oracle, estimate $|\mathrm{perm}\,(M)|^2$

Reductions

- Compute permanent in $\mathrm{BPP}^{\mathrm{NP}}$

Perm is #P-complete

- $\mathrm{P}^{\#\mathrm{P}}$ lies within $\mathrm{BPP}^{\mathrm{NP}}$

Toda's Theorem

- Polynomial hierarchy collapses

Approximate Bosons are Hard: Proof

- Classically *approximately* simulate identical bosons

Approx counting

- Using NP oracle, estimate $|\text{perm}(M)|^2$ *of random M with high probability*

Random self-reducibility + conjectures

- Compute permanent in BPP^{NP}

Perm is #P-complete

- $\text{P}^{\#\text{P}}$ lies within BPP^{NP}

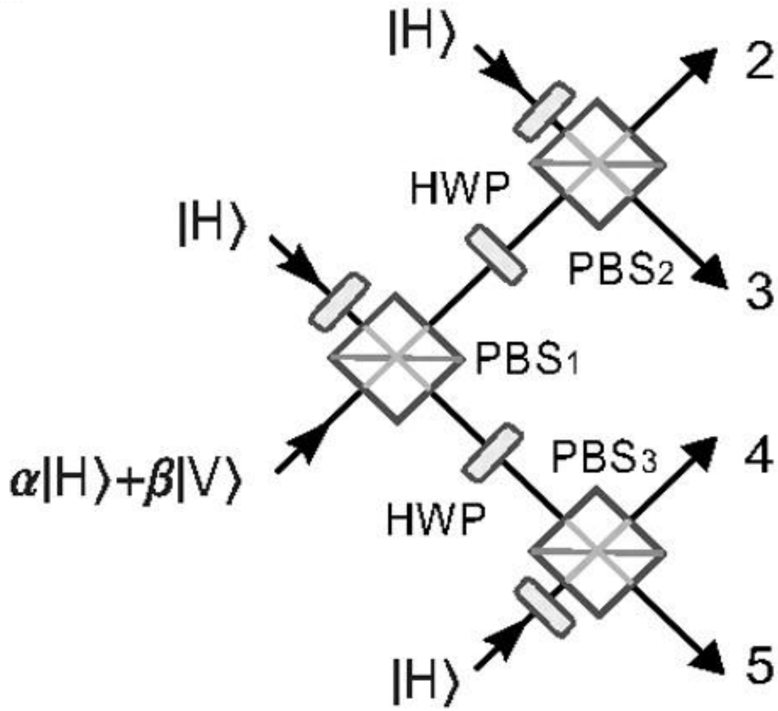
Toda's Theorem

- Polynomial hierarchy collapses

Experimental Prospects

Linear Optics

- Photons and half-silvered mirrors



- Beamsplitters + phaseshifters are universal

Challenge: Do These Reliably

- Encode values into mirrors
- Generate single photons
- Have photons hit mirrors at same time
- Detect output photons

Proposed Experiment

- Use m=20, n=10
- Choose U at random
- Check by brute force!