Quantum simulation and analogue computation

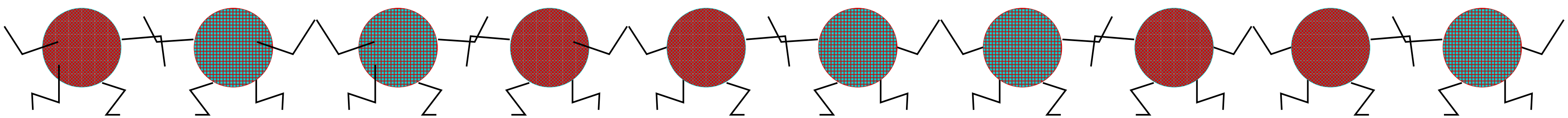
Viv Kendon

Rob Wagner, Mark S. Everitt, Martin L. Jones

Talk at **CRPC workshop, Oxford** *Monday 24th August 2009*

Quantum Information
School of Physics
& Astronomy
University of Leeds
Leeds LS2 9JT
V.Kendon@leeds.ac.uk

Funding: Royal Society University Research Fellowship
EPSRC (MSE & RW)



Overview

1. Introduction *(key features of quantum computing)*

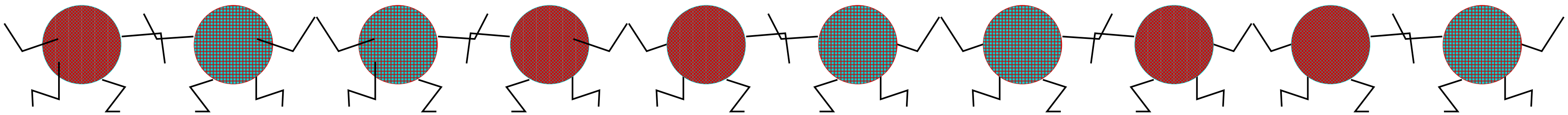
2. Quantum simulation *(simulating one quantum system using another)*

3. Analogue computing *(Shannon's GPAC)*

4. Continuous variable quantum computing *(quantum version of analogue computing)*

5. How many qubits do we need to make a useful quantum computer?

6. Outlook *Are we going to get one any time soon?*



Introduction

Quantum Information is built from the idea that:

Quantum Logic allows greater efficiency than **Classical Logic**

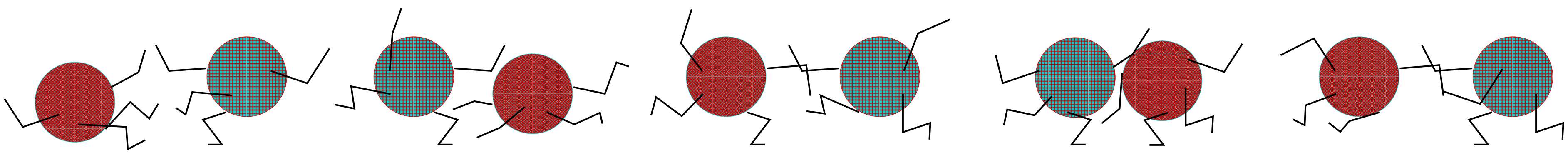
- needs some justification: *how might translate into better computing devices?*
- depends on definition of **EFFICIENCY**

  in theory: polynomial scaling with system size

  in practice: produces answers on human timescales

Quadratic improvement exploits quantum coherence, interference effects

Exponential speed up by exploiting parallelism in quantum superposition



Programming a quantum computer

Generic types of quantum algorithms:

- Simulation of quantum systems (Feynman 1985) [exponential]
- Promise problems – e.g. Deutsch-Jozsa (1992) [oracle]
- Quantum Fourier transform – e.g. Shor's factoring algorithm (1994) [exponential]
- Grover's search of unsorted database (1996) [quadratic]
- Quantum versions of random walks (2002) [quadratic; exponential with oracle]

Many variants on basic types – *also, quantum game theory, quantum neural nets...*

Most other quantum information processing is based on

- communications protocols... [factor of 2 + shared randomness]

Physical systems vs algorithms

Be clear about difference between physical systems and algorithms:

Examples of Random Walks:

	quantum	*classical*
physical	particle in optical lattice	snakes and ladders (board game)
computer	glued trees algorithm	lattice QCD calculation

- Can also do classical computer simulation of all of these four possibilities!

[VK, Phil Trans Roy Soc A, 364, 3407--3422 (2006)]

...try to keep these multiple levels of abstraction clear...

Unary vs Binary Coding

Number	Unary	Binary
0		0
1	●	1
2	●●	10
3	● ● ●	11
4	● ● ●●	100
...	...	...
N	$N \times \bullet$	$\log_2 N$ bits

Read out:

Unary: distinguish between measurements with N outcomes

Binary: $\log_2 N$ measurements with 2 outcomes each

$\longrightarrow$ exponentially better for accuracy

[Ekert & Jozsa PTRSA 356 1769--82 (1998)]

binary encoding $\longrightarrow$ exponential gain (reduction) in size of memory over unary

[does not have to be binary: Blume-Kohout, Caves, I. Deutsch Found. Phys. 32 1641-1670 quant-ph/0204157]

Quantum Simulation

A quantum system can simulate another quantum system efficiently

[Lloyd Science 273, 1073 1996] – map one Hilbert space directly onto the other

– Trotter approximation for unitary evolution using Lie product formula or variations:

$$\exp\{iHt\} \simeq (\exp\{iH_1 t/n\} \exp\{iH_2 t/n\} \ldots \exp\{iH_m t/n\})^n + O(t^2/n)[H_j, H_k]$$

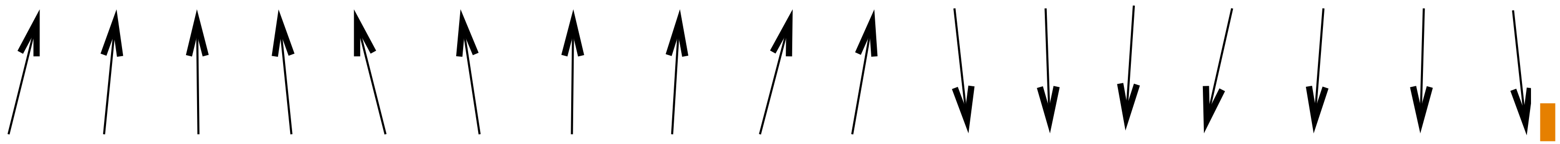
$$H = H_1 + H_2 + \cdots + H_m$$

Has been demonstrated [Somaroo et al., 1999], using NMR quantum computers

However, because no binary encoding... accuracy is a problem

...does not scale efficiently with time needed to run simulation

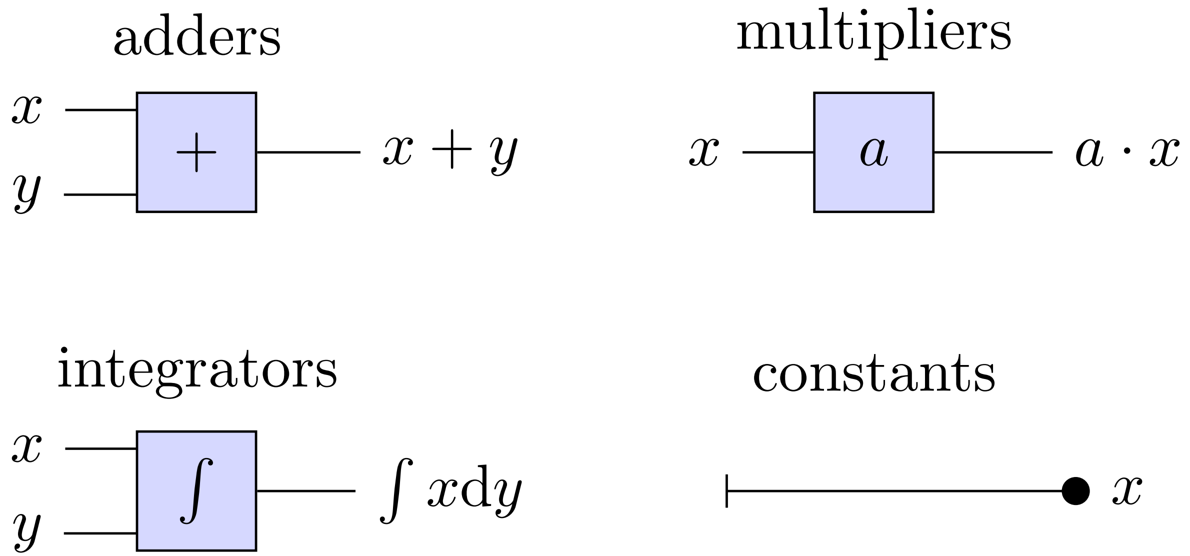
[Ken Brown et al. quant-ph/0601021]



Analogue Computing

Qantum simulation is like *analogue computation*:

- encode numbers into size of some continuous quantity such as height of a water column or electrical voltage
- form circuit from small set of components, e.g., Shannon's GPAC elements



GPAC can solve any ordinary differential equation – extensions can do more functions

exponential scaling: one extra bit of precision requires double the resources

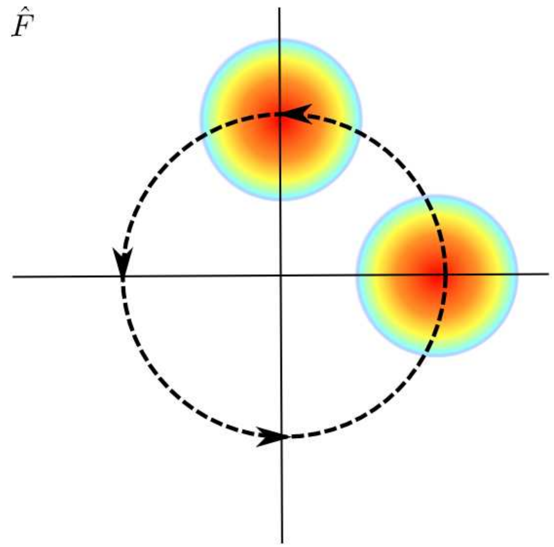
Continuous Variable Quantum Computing

– quantum version of analogue computing!

uncertainty relations: don't have well-defined continuous quantities

– use infinitely squeezed states in theory

– more practical: Gaussian states *[Lloyd + Braunstein quant-ph/9810082v1]*



Universal set of operations, similar to GPAC:

- Displacements
- Fourier Transform
- Single mode squeezing
- Two-mode squeezing
- nonlinearity (at least cubic)

enough to construct any polynomial in variables

CVQC in a micro maser

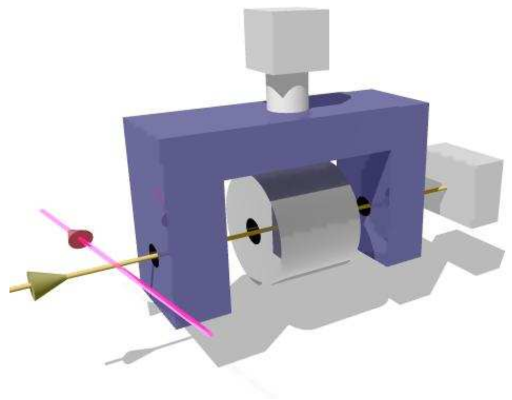
Physical implementation of CVQC: trap light in a cavity and control with atoms

Jaynes-Cummings model system:

– one of the simplest and “cleanest” is the micro maser (one atom maser)

Practical universal set of operations, needs some modification:

- Displacements (easy)
- Fourier Transform (very easy)
- Single mode squeezing (OK)
- Two-mode squeezing (a bit trickier, use single mode + interaction)
- nonlinearity, at least cubic (the hard bit!)
- read out (measurement, a bit tricky)



(work with PhD student Rob Wagner on implementation in micromaser)

Computing something we can't classically...

Simulating a quantum system: example – $N \times$ 2-state particles

$\longrightarrow 2^N$ possible states – could be in superposition of all of them

classical requires:

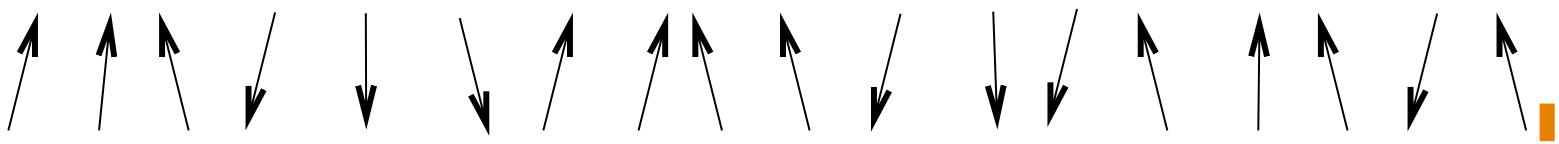
one complex number per state: $2^{N+1} \times$size-of-double $\longrightarrow$ 1Gbyte holds $N = 26$

record: N=36 in 1Terabyte – each additional particle doubles memory required!

`[De Raedt et al, quant-ph/0608239]`

more than 40 or so qubits = beyond classical limit

(note: may not need all superpositions, e.g., if only nearest neighbour interactions, so larger classical simulations possible...many papers on subject)



Computing something we can't classically...

Shor's factoring algorithm:

need to beat: best classical to date: 200 digits (RSA-200) = approx 665 bits

Shor's quantum algorithm needs: $2n$ qubits in QFT register plus $5n$ qubits for modular exponentiation = $7n$ logical qubits – 665 bit number needs 4655 logical qubits

now add error correction: depends on error rates...

if error rate close to threshold of 10^{-3} to 10^{-4}, need more error correction

(note: threshold error rate is smaller than any experiment has achieved)

for low error rates, maybe 20–200 physical qubits per logical qubit

for high error rates, blows up quickly, maybe 10^5 per logical qubit

suggests we may need Teraqubit quantum computers to break factoring

– scaling favours quantum, but the crossover point is high

Outlook

- 25 years since Feynman + Deutsch first introduced idea of quantum computing
- 15 years since Shor's factoring algorithm
- still only have toy quantum computers, no front runner on architecture

*if first useful application is **quantum simulation** (Feynman's original idea), then*

- resources, scaling, error correction all different from digital
- room for more radical ideas from analogue computing
- gaps in theory of analogue – quantum and classical – to be filled