Information Theory Meets Quantum Physics

The magic of wave dynamics

Apoorva Patel

Centre for High Energy Physics and

Supercomputer Education and Research Centre

Indian Institute of Science, Bangalore

7 September 2012, IEEE-IISc Information Theory Workshop, Bangalore

Computation

Any physical process evolves an initial state to a final state under the influence of certain interactions.

Initial state $\longrightarrow$ Interactions $\longrightarrow$ Final state

Computation

Any physical process evolves an initial state to a final state under the influence of certain interactions.

Initial state $\longrightarrow$ Interactions $\longrightarrow$ Final state

A computer processes the given input to an output according to specified instructions.

Input $\longrightarrow$ Processing $\longrightarrow$ Output

Computation

Any physical process evolves an initial state to a final state under the influence of certain interactions.

Initial state $\longrightarrow$ Interactions $\longrightarrow$ Final state

A computer processes the given input to an output according to specified instructions.

Input $\longrightarrow$ Processing $\longrightarrow$ Output

It purposefully processes meaningful information.
The meaning arises from mapping physical properties (hardware) to mathematical terminology (software).

Computation

Any physical process evolves an initial state to a final state under the influence of certain interactions.

Initial state $\longrightarrow$ Interactions $\longrightarrow$ Final state

A computer processes the given input to an output according to specified instructions.

Input $\longrightarrow$ Processing $\longrightarrow$ Output

It purposefully processes meaningful information.
The meaning arises from mapping physical properties (hardware) to mathematical terminology (software).

The availability of different physical interactions makes it possible to design different types of computers.

Types of Computers

The efficiency and stability of a computational paradigm depend on the nature of implemented physical interactions.

Classical (particle dynamics):
Discrete Boolean logic is implemented using digital circuits.

Types of Computers

The efficiency and stability of a computational paradigm depend on the nature of implemented physical interactions.

Classical (particle dynamics):
Discrete Boolean logic is implemented using digital circuits.
Wave (classical wave dynamics):
Analog variables can superpose, interfere, disperse etc., and are convenient for differentiation and integration.
Waves have been widely used in communications, and in simple analog computers (e.g. RLC circuits), but they have been left out of digital computation.

Types of Computers

The efficiency and stability of a computational paradigm depend on the nature of implemented physical interactions.

Classical (particle dynamics):
Discrete Boolean logic is implemented using digital circuits.
Wave (classical wave dynamics):
Analog variables can superpose, interfere, disperse etc., and are convenient for differentiation and integration.
Waves have been widely used in communications, and in simple analog computers (e.g. RLC circuits), but they have been left out of digital computation.
Quantum (particle+wave dynamics):
Unitary evolution is implemented using quantum states. Combination of particle and wave properties produces unusual correlations called entanglement.

Types of Computers

The efficiency and stability of a computational paradigm depend on the nature of implemented physical interactions.

Classical (particle dynamics):
Discrete Boolean logic is implemented using digital circuits.
Wave (classical wave dynamics):
Analog variables can superpose, interfere, disperse etc., and are convenient for differentiation and integration.
Waves have been widely used in communications, and in simple analog computers (e.g. RLC circuits), but they have been left out of digital computation.
Quantum (particle+wave dynamics):
Unitary evolution is implemented using quantum states. Combination of particle and wave properties produces unusual correlations called entanglement.

Wave algorithms can be useful in situations where spatial resources are cheap and quantum algorithms are fragile.

What is a quantum computer?

It is a computer whose elementary hardware components work according to the laws of quantum mechanics.
(The hardware components of classical digital computers work according to the laws of electronic circuits.)

What is a quantum computer?

It is a computer whose elementary hardware components work according to the laws of quantum mechanics.
(The hardware components of classical digital computers work according to the laws of electronic circuits.)

There is a lot more to computation than Boolean algebra!
In going from classical to quantum computers, the concept of what is computable and what is not does not change, but the criteria of computational efficiency do.

What is a quantum computer?

It is a computer whose elementary hardware components work according to the laws of quantum mechanics.
(The hardware components of classical digital computers work according to the laws of electronic circuits.)

There is a lot more to computation than Boolean algebra!
In going from classical to quantum computers, the concept of what is computable and what is not does not change, but the criteria of computational efficiency do.

Present Status

Laws of quantum mechanics are precisely known.
Theoretical foundation of the subject is clear.
Elementary hardware components work as predicted.

What is a quantum computer?

It is a computer whose elementary hardware components work according to the laws of quantum mechanics.
(The hardware components of classical digital computers work according to the laws of electronic circuits.)

There is a lot more to computation than Boolean algebra!
In going from classical to quantum computers, the concept of what is computable and what is not does not change, but the criteria of computational efficiency do.

Present Status

Laws of quantum mechanics are precisely known.
Theoretical foundation of the subject is clear.
Elementary hardware components work as predicted.
Large scale integration (say 10 or more components) is a technological challenge. Noone knows when that will arrive, or what a quantum computer will be used for.

Another Look at Computation

Computation is Processing and Communication of Semantic Information expressed using a Language.

One finds examples of many types of information processing systems in the physical world.

How can a general information theory be systematically developed to cover all types of computational schemes?

How can one design the optimal computer for a given task?

Another Look at Computation

Computation is Processing and Communication of Semantic Information expressed using a Language.

One finds examples of many types of information processing systems in the physical world.

How can a general information theory be systematically developed to cover all types of computational schemes?

How can one design the optimal computer for a given task?

Optimisation

Minimise physical resources (space, time, energy, . . .).
Software: What is the task? What is the algorithm?
Hardware: How are the operations implemented?
Efficiency of information processing depends on both software and hardware.

Technical Terms

Data: They describe a particular realisation of the physical system, amongst its many possible states.

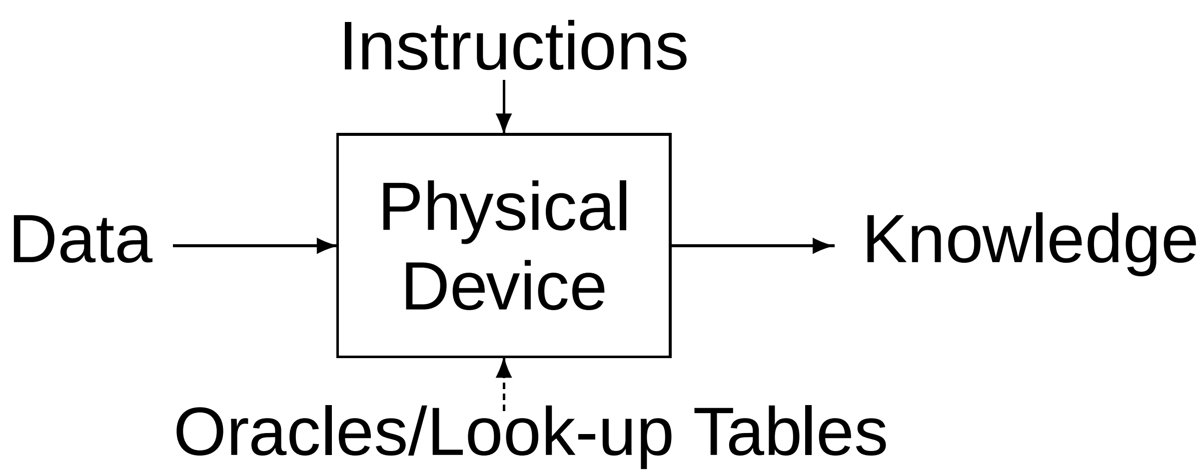
Information: It is the abstract mathematical property obtained by detaching all the physical characteristics from data.

Knowledge: It is obtained by adding a sense of purpose to the abstract information.

Technical Terms

Data: They describe a particular realisation of the physical system, amongst its many possible states.
Information: It is the abstract mathematical property obtained by detaching all the physical characteristics from data.
Knowledge: It is obtained by adding a sense of purpose to the abstract information.

Information = Data - Physical Realisation
Knowledge = Information + Interpretation



Importance of Physics

1. Abstract information can be manipulated with precise mathematical rules, without going into nitty-gritty of its origin or meaning.
2. The manipulations can only be implemented using physical devices.
3. The interpretation of a language has to be established through physical properties.

Importance of Physics

1. Abstract information can be manipulated with precise mathematical rules, without going into nitty-gritty of its origin or meaning.
2. The manipulations can only be implemented using physical devices.
3. The interpretation of a language has to be established through physical properties.

Abstract information theory does not tell us what physical realisation would be appropriate for a particular message, nor does it tell us the best way of implementing a computational task.

These choices have to be made by analysing the type (and not the amount) of information, and inspecting the available physical resources.

Hierarchical Processing of Information

Computers		Living organisms
Data	Input	Environmental signals
Pre-processor	High level	Sense organs
Compiler	$\Uparrow$	Nervous system
Assembler	Translation	Brain
	$\Downarrow$	
Machine code	Low level	Electrochemical signals
Electrical signals	Execution	Proteins
Programmer	Programme	DNA

Hierarchical Processing of Information

Computers		Living organisms
Data	Input	Environmental signals
Pre-processor	High level	Sense organs
Compiler	$\Uparrow$	Nervous system
Assembler	Translation	Brain
	$\Downarrow$	
Machine code	Low level	Electrochemical signals
Electrical signals	Execution	Proteins
Programmer	Programme	DNA

High level processing is abstract, with a wide variety of instructions and subjective adaptations.
Low level processing is directly related to physical properties, with a limited number of instructions and tasks.

At the lowest level of information processing, the physical objects that carry the message have to convey the information as well as its interpretation.

Optimisation Criteria

Minimise errors: Unwanted noise must be separated from genuine transformations. So use clearly distinguishable building blocks, with discrete operations. $\Longrightarrow$ Digitisation

Optimisation Criteria

Minimise errors: Unwanted noise must be separated from genuine transformations. So use clearly distinguishable building blocks, with discrete operations. $\Longrightarrow$ Digitisation

1. Discrete values are mapped to non-overlapping regions of physical properties. Fault tolerance is achieved by ignoring small variations as noise, and by considering large changes to be genuine transformations.

Optimisation Criteria

Minimise errors: Unwanted noise must be separated from genuine transformations. So use clearly distinguishable building blocks, with discrete operations. $\Longrightarrow$ Digitisation

1. Discrete values are mapped to non-overlapping regions of physical properties. Fault tolerance is achieved by ignoring small variations as noise, and by considering large changes to be genuine transformations.

2. Practical applications need only bounded error calculations, i.e. non-zero tolerance level for continuous variables (or finite success rate for probabilistic results).

3. Polynomial increase in the number of discrete variables (or trials) can provide exponential increase in precision, e.g. the place value system for numbers.

. . . contd.

Minimise resources: Use a small number of building blocks, with simple and quick operations. Join them together in as many different ways as possible. $\Longrightarrow$ Versatile composition

Minimise resources: Use a small number of building blocks, with simple and quick operations. Join them together in as many different ways as possible. $\Longrightarrow$ Versatile composition

1. Miniaturisation (with stability) reduces resources.
2. Inefficiently encoded data can be compressed.
   Our written languages have enough redundancy,
   but our number system has maximum versatility.

Minimise resources: Use a small number of building blocks, with simple and quick operations. Join them together in as many different ways as possible. $\Longrightarrow$ Versatile composition

1. Miniaturisation (with stability) reduces resources.
2. Inefficiently encoded data can be compressed.
   Our written languages have enough redundancy,
   but our number system has maximum versatility.

These criteria often confilct, and trade-offs are necessary.
Example: Binary variables are processed in the electrical form (in the CPU) for high speed, and are stored in the magnetic form (on the disk) for high stability.

Physical hardware properties govern the expression and the grammar of the language.

Minimise resources: Use a small number of building blocks, with simple and quick operations. Join them together in as many different ways as possible. $\Longrightarrow$ Versatile composition

1. Miniaturisation (with stability) reduces resources.
2. Inefficiently encoded data can be compressed.
   Our written languages have enough redundancy,
   but our number system has maximum versatility.

These criteria often confilct, and trade-offs are necessary.
Example: Binary variables are processed in the electrical form (in the CPU) for high speed, and are stored in the magnetic form (on the disk) for high stability.

Physical hardware properties govern the expression and the grammar of the language.

Nevertheless, the language with the smallest set of building blocks (for a given task) has a unique status in the optimisation procedure.

Minimal Language

- Largest tolerance against errors.

(Discrete variables are spread as far apart as possible in the available range of physical hardware properties.)

- Smallest instruction set.

(Number of possible transformations is limited.)

- High density of packing and quick operations.

(These more than make up for the increased depth of computation.)

- Simplest language, without need of translation.

(Simple physical responses of the hardware can be used.)

Minimal Language

- Largest tolerance against errors.

(Discrete variables are spread as far apart as possible in the available range of physical hardware properties.)

- Smallest instruction set.

(Number of possible transformations is limited.)

- High density of packing and quick operations.

(These more than make up for the increased depth of computation.)

- Simplest language, without need of translation.

(Simple physical responses of the hardware can be used.)

Boolean algebra provides the minimal classical language for encoding information as 1-dimensional sequences.

General Computational Framework

Generalise the concept of a language, from a "sequence of letters" to a "collection of building blocks".

General Computational Framework

Generalise the concept of a language, from a "sequence of letters" to a "collection of building blocks".

Collections: The building blocks can be arranged in the space-time in many different ways.
Building blocks: Physical properties (generically encoded using groups) express the meaning of the building blocks.
Processing: Allowed changes in the properties of building blocks exhaust the possible manipulations of information.

Group representations fix the structure of the language, and group transformations provide the rules for processing information.

. . . contd.

For a given task, the smallest discrete group that can implement it is the ideal candidate for the optimal language.

In digitisation of continuous spaces (of physical properties), a discrete state is associated not with just a point on the manifold but with a finite neighbourhood of a point.

For a given task, the smallest discrete group that can implement it is the ideal candidate for the optimal language.

In digitisation of continuous spaces (of physical properties), a discrete state is associated not with just a point on the manifold but with a finite neighbourhood of a point.

For a d-dim space, the simplest building block is a simplex, i.e. a set of $(d+1)$ points.

The dimension of a group is the number of its generators. In the dual description, the minimal building block set is the d-dim fundamental representation and the 1-dim identity representation.

For a given task, the smallest discrete group that can implement it is the ideal candidate for the optimal language.

In digitisation of continuous spaces (of physical properties), a discrete state is associated not with just a point on the manifold but with a finite neighbourhood of a point.

For a d-dim space, the simplest building block is a simplex, i.e. a set of $(d+1)$ points.

The dimension of a group is the number of its generators. In the dual description, the minimal building block set is the d-dim fundamental representation and the 1-dim identity representation.

In the smallest discrete realisation, the entire group is replaced by a single simplex. A collection of simplices can then represent any quantity to the desired precision (e.g. the place value system for numbers).

Types of Collections

$0 - dim$: Multiple signals at the same point in space and time, i.e. superposition. Different states of internal degrees of freedom encode different signals. Only one signal can be extracted at a time.

Examples: Radio broadcasts, Cellular phones.

Types of Collections

$0 - dim$: Multiple signals at the same point in space and time, i.e. superposition. Different states of internal degrees of freedom encode different signals. Only one signal can be extracted at a time.
Examples: Radio broadcasts, Cellular phones.

$1 - dim$: Building blocks arranged as an ordered sequence. This tensor product structure is routinely used in conventional information theory.
Examples: Written languages (sequence in space),
Spoken languages (sequence in time).

Types of Collections

$0 - dim$: Multiple signals at the same point in space and time, i.e. superposition. Different states of internal degrees of freedom encode different signals. Only one signal can be extracted at a time.
Examples: Radio broadcasts, Cellular phones.

$1 - dim$: Building blocks arranged as an ordered sequence. This tensor product structure is routinely used in conventional information theory.
Examples: Written languages (sequence in space),
Spoken languages (sequence in time).

$2 - dim$: Combination of multiple ordered sequences. Information can reside in correlations amongst sequences without being present in any individual sequence.
Examples: Eyes and ears, Space-time codes,
Parallax and gradient detection.

Types of Collections (contd.)

$3 - dim$: Building blocks encode structural information, for establishing lock and key mechanisms.
Examples: Proteins and other biomolecules.

Types of Collections (contd.)

$3 - dim$: Building blocks encode structural information, for establishing lock and key mechanisms.
Examples: Proteins and other biomolecules.

$4 - dim$: Complete description of any space-time event in our universe.
Example: Global positioning system (GPS).

Types of Collections (contd.)

$3 - dim$: Building blocks encode structural information, for establishing lock and key mechanisms.
Examples: Proteins and other biomolecules.

$4 - dim$: Complete description of any space-time event in our universe.
Example: Global positioning system (GPS).

Simultaneous use of multiple dimensionalities is also possible, with building blocks having multiple properties!
Examples: Quantum computers (0 and 1 dim),
Proteins (1 and 3 dim).

Types of Collections (contd.)

$3 - dim$: Building blocks encode structural information, for establishing lock and key mechanisms.
Examples: Proteins and other biomolecules.

$4 - dim$: Complete description of any space-time event in our universe.
Example: Global positioning system (GPS).

Simultaneous use of multiple dimensionalities is also possible, with building blocks having multiple properties!
Examples: Quantum computers (0 and 1 dim),
Proteins (1 and 3 dim).

Particle features: Countability, Order, Number density, Shape, Structure.
Wave features: Superposition, Differential analysis (parallax), Interference (multiple paths).

Types of Building Blocks

A combination of properties, in both internal and external space, can be used to describe distinguishable objects.

Types of Building Blocks

A combination of properties, in both internal and external space, can be used to describe distinguishable objects.

$1 - dim$: Groups with a single generator include cyclic groups, integers and the real line.
The minimal simplex is $Z_2 = \{0, 1\}$.
The binary language can be extended to a d-dim situation, as the Cartesian product $(Z_2)^d$, and is therefore convenient as a general purpose language.

Types of Building Blocks

A combination of properties, in both internal and external space, can be used to describe distinguishable objects.

$1-dim$: Groups with a single generator include cyclic groups, integers and the real line.
The minimal simplex is $Z_2 = \{0, 1\}$.
The binary language can be extended to a d-dim situation, as the Cartesian product $(Z_2)^d$, and is therefore convenient as a general purpose language.

$2-dim$: The simplex is a triangle. Triangulation (or its dual hexagonal form) is useful for discrete description of arbitrary surfaces. Graphene structure may become useful in atomic scale lithography.

Types of Building Blocks (contd.)

$3 - dim$: The simplex is a tetrahedron. At molecular scale, sp^3-hybridised orbitals provide its dual form. Tetrahedral geometry of carbon provides a convenient way to understand the 3-dim language of proteins.

Types of Building Blocks (contd.)

$3 - dim$: The simplex is a tetrahedron. At molecular scale, sp^3-hybridised orbitals provide its dual form. Tetrahedral geometry of carbon provides a convenient way to understand the 3-dim language of proteins.

$SU(2)$: Description of quantum bits is based on this group with three generators. Arbitrary states of a qubit (including mixed states) can be fully described using a density matrix, which is a linear combination of the four operators $\{1, \sigma_x, \sigma_y, \sigma_z\}$. (Four real numbers $\equiv$ Two complex numbers)

Types of Building Blocks (contd.)

$3-dim$: The simplex is a tetrahedron. At molecular scale, sp^3-hybridised orbitals provide its dual form. Tetrahedral geometry of carbon provides a convenient way to understand the 3-dim language of proteins.

$SU(2)$: Description of quantum bits is based on this group with three generators. Arbitrary states of a qubit (including mixed states) can be fully described using a density matrix, which is a linear combination of the four operators $\{1, \sigma_x, \sigma_y, \sigma_z\}$. (Four real numbers $\equiv$ Two complex numbers)

Larger groups have been used in error correcting codes and cryptography, but not for processing information.

Types of Processing

$0 - dim$: Only mathematical operation allowed with superposition is addition.
It is commutative and produces interference effects.

Types of Processing

$0 - dim$: Only mathematical operation allowed with superposition is addition.
It is commutative and produces interference effects.

$1 - dim$: Z_2 is a field—the smallest one. It allows two different commutative operations, addition and multiplication, which are the basis of all our arithmetic.

Types of Processing

$0 - dim$: Only mathematical operation allowed with superposition is addition.
It is commutative and produces interference effects.

$1 - dim$: Z_2 is a field—the smallest one. It allows two different commutative operations, addition and multiplication, which are the basis of all our arithmetic.

$d > 1$: In higher dimensions, addition generalises to translation, and multiplication to scale transformation.
But rotations appear as well (non-commutative for $d > 2$).
Algebra generated by lattice transformations is much more powerful than common arithmetic.

More and more group operations become possible with increasing dimensionality.

Database Search

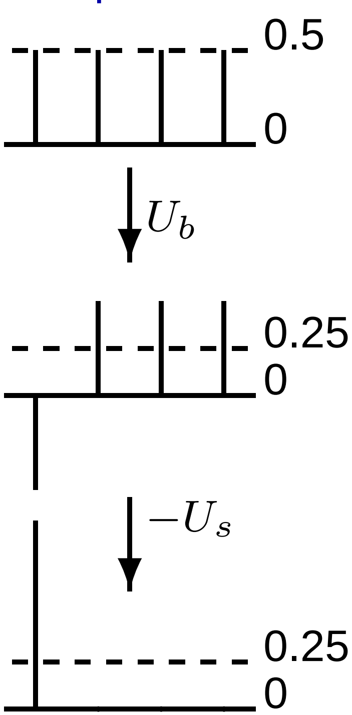
Classical:
Binary tree search is the optimal classical algorithm.
A sorted database of N items can be searched using $\log_2 N$ binary questions.
An unsorted database of N items can be searched using $N/2$ binary questions with memory, and using N binary questions without memory.

Quantum/Wave:
Wave mechanics works with amplitudes and not with probabilities. Superposition of amplitudes can yield constructive as well as destructive interference.
Optimal search solutions differ from the classical ones.

Grover's algorithm: An unsorted database of N items can be optimally searched using $(\pi/4)\sqrt{N}$ binary questions.

Grover's Database Search

The steps of the algorithm for the simplest case of 4 items in the database. Let the first item be desired by the oracle.



	Amplitudes	Algorithmic Steps	Physical Implementation
(1)		Uniform distribution	Equilibrium configuration
	U_b	Quantum oracle	Binary question
(2)		Amplitude of desired state flipped in sign	Sudden perturbation
	$-U_s$	Reflection about average	Overrelaxation
(3)		Desired state reached	Opposite end of oscillation
(4)	Projection	Algorithm is stopped	Measurement

(Dashed line denotes the average amplitude.)

Wave Algorithms

In quantum systems, $|A|^2$ gives the probability of a state, which is transfered from the initial state to the final one.

In wave systems, $|A|^2$ gives the energy of a mode, which is transfered from the initial configuration to the final one.

Wave Algorithms

In quantum systems, $|A|^2$ gives the probability of a state, which is transfered from the initial state to the final one.

In wave systems, $|A|^2$ gives the energy of a mode, which is transfered from the initial configuration to the final one.

Wave systems with coupled vibrational (and rotational) modes, and small damping, can be made easily.

Efficient schemes to transfer/redistribute energy can have many practical applications—from mechanical systems to chemical, electrical and biological ones.

Wave Algorithms

In quantum systems, $|A|^2$ gives the probability of a state, which is transfered from the initial state to the final one.

In wave systems, $|A|^2$ gives the energy of a mode, which is transfered from the initial configuration to the final one.

Wave systems with coupled vibrational (and rotational) modes, and small damping, can be made easily.

Efficient schemes to transfer/redistribute energy can have many practical applications—from mechanical systems to chemical, electrical and biological ones.

Video of amplitude amplification in coupled oscillators

Possible uses

Focusing of energy:

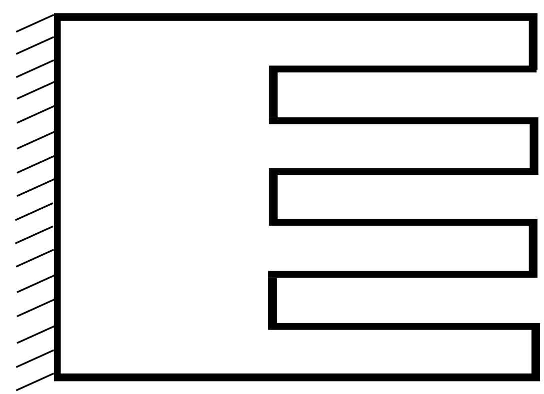
Concentration of total energy of a coupled oscillator system into a specific oscillator can have potential applications in processes that are highly sensitive to energy availability.

Possible uses

Focusing of energy:
Concentration of total energy of a coupled oscillator system into a specific oscillator can have potential applications in processes that are highly sensitive to energy availability.

Nanomechanical devices: A coupled oscillator system can provide efficient focusing of energy at a specific location, when one cannot directly control the component concerned.

For example, a comb-shaped cantilever beam can be used as a selective switch or a sensor.



Catalysis: There exist many processes that need crossing of an energy threshold for completion. Their rates are typically governed by the Boltzmann factor for the energy barrier, $\exp(-E_{\rm barrier}/kT)$. Energy amplification can speed up the rates of such processes dramatically.

Catalysis: There exist many processes that need crossing of an energy threshold for completion. Their rates are typically governed by the Boltzmann factor for the energy barrier, $\exp(-E_{\rm barrier}/kT)$. Energy amplification can speed up the rates of such processes dramatically.

Dispersal of energy:

Grover's algorithm is fully reversible.
Running it backwards, distributes large initial energy in one of the oscillators equally among its partners.

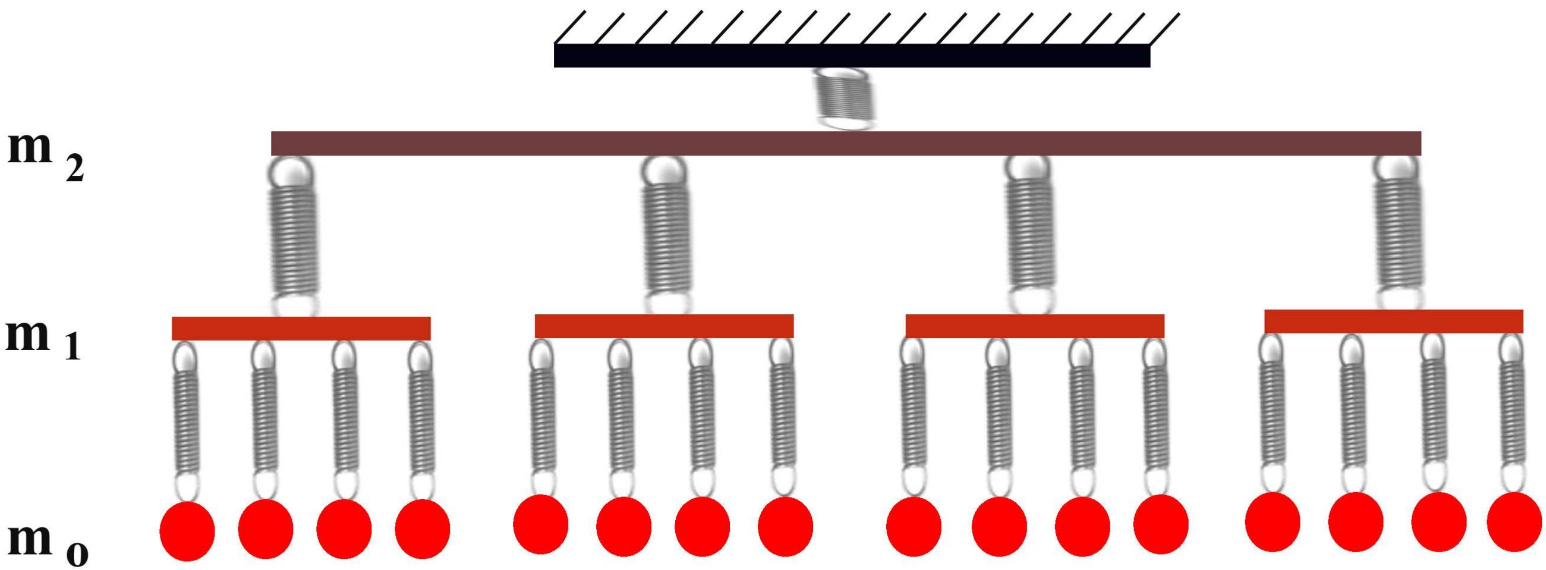
Catalysis: There exist many processes that need crossing of an energy threshold for completion. Their rates are typically governed by the Boltzmann factor for the energy barrier, $\exp(-E_{\rm barrier}/kT)$. Energy amplification can speed up the rates of such processes dramatically.

Dispersal of energy:

Grover's algorithm is fully reversible.
Running it backwards, distributes large initial energy in one of the oscillators equally among its partners.

Shock absorbers and vibrational isolation: Instead of damping a single perturbed oscillator, it is much more efficient to disperse the energy into several oscillators while damping them together.

A hierarchical system of oscillators—four small ones coupled to a big one at every level with appropriate mass, spring and damping parameters—can provide a practical realisation of this idea.



(The initial impulse is taken to be a local disturbance, which subsequently spreads out.)

Language of the Genes

1. What is the information processing task carried out by the genetic machinery?
   Assembling molecules by picking up components from an unsorted database.

Language of the Genes

1. What is the information processing task carried out by the genetic machinery?
   Assembling molecules by picking up components from an unsorted database.
2. What is the optimal way of carrying out this task?
   Lov Grover's quantum search algorithm.
   (Requires wave dynamics.)

Language of the Genes

1. What is the information processing task carried out by the genetic machinery?
   Assembling molecules by picking up components from an unsorted database.
2. What is the optimal way of carrying out this task?
   Lov Grover's quantum search algorithm.
   (Requires wave dynamics.)
3. Classically two nucleotide bases (one complementary pair) are sufficient to encode the genetic information. That would be a simpler system, and so it would have preceded (during evolution) the four nucleotide base system found in nature.
   Was the advantage provided by the wave algorithm the real incentive for nature to complicate the system?

Outlook

The paradigm of computation can be vastly expanded beyond mere arithmetic. Explore the possibilities!

Outlook

The paradigm of computation can be vastly expanded beyond mere arithmetic. Explore the possibilities!

User: Looks at computers from the top level down, from abstract mathematical operations to the transistors embedded in silicon chips.

Designer: Explores efficient computers from the bottom level up, from the elementary building blocks to the complicated languages.

Outlook

The paradigm of computation can be vastly expanded beyond mere arithmetic. Explore the possibilities!

User: Looks at computers from the top level down, from abstract mathematical operations to the transistors embedded in silicon chips.

Designer: Explores efficient computers from the bottom level up, from the elementary building blocks to the complicated languages.

Biological systems have evolved from the bottom level up, and by trial and error discovered many computational systems over billions of years.

Life is based on a nanotechnology that works—has worked for billions of years. There is a lot to learn from that!

Summary

There is a lot to explore, in both hardware and software!

Hardware: Find methods to deduce optimal physical realisation of building blocks for a given computational task.

Software: Develop group theoretical techniques to construct directly implementable high level instructions.

Wisdom: Learn from biological examples.
Digitally punctuated analog computation, stabilised by feedback, greatly improves energy efficiency.

References

All my papers are available at http://arXiv.org/

Minimal Languages:

quant-ph/0503068: The Future of Computation,
Proceedings of QICC 2005, Kharagpur, Allied Publishers (2006) 197-206

0705.3895[q-bio.GN]: Towards Understanding the Origin of Genetic Languages,
Chapter 10 in “Quantum Aspects of Life",
Eds. D. Abbott et al., Imperial College Press (2008) 187-219

Wave Algorithms:

quant-ph/0609042: Wave Algorithms: Optimal Database Search and Catalysis,
AIP Conference Proceedings 864 (2006) 261-272

0711.4733[quant-ph]: A Coupled Oscillator Model
for Grover’s Quantum Database Search Algorithm,
Intel International Science and Engineering Fair, Albuquerque, USA (2007)

1108.1659[quant-ph]: Quantum Computation: Particle and Wave Aspects of Algorithms,
Resonance 16 (2011) 821-835

Some Directions

The number of fundamental physical interactions is small.
A variety of computational schemes can still be created,
by different combinations of the fundamental ingredients.

(e.g. all of condensed matter physics is essentially the consequence of electrodynamics.)

Some Directions

The number of fundamental physical interactions is small. A variety of computational schemes can still be created, by different combinations of the fundamental ingredients.

(e.g. all of condensed matter physics is essentially the consequence of electrodynamics.)

1. Operations of calculus (e.g. differentiation, integration) are easier to carry out using continuous variables instead of discrete ones. Analogue computation, punctuated by digitisation to control errors, can be convenient and also energy efficient.

Some Directions

The number of fundamental physical interactions is small. A variety of computational schemes can still be created, by different combinations of the fundamental ingredients.

(e.g. all of condensed matter physics is essentially the consequence of electrodynamics.)

1. Operations of calculus (e.g. differentiation, integration) are easier to carry out using continuous variables instead of discrete ones. Analogue computation, punctuated by digitisation to control errors, can be convenient and also energy efficient.

2. The depth of computation can be reduced by direct execution of high level instructions. This can be achieved using special purpose components and configurable systems.
   Biology has excelled in creating such systems.

. . . contd.

Some Directions (contd.)

3. Can fractal arrangement of building blocks be used in some new type of information processing? (Self-similar patterns occur in concatenated error correcting codes.)

Some Directions (contd.)

3. Can fractal arrangement of building blocks be used in some new type of information processing? (Self-similar patterns occur in concatenated error correcting codes.)

4. Building blocks having multiple physical properties, each described by a particular group, can cut down resources by simultaneous execution of multiple transformations. (e.g. electron has position, spin, energy level, etc.)

Some Directions (contd.)

3. Can fractal arrangement of building blocks be used in some new type of information processing? (Self-similar patterns occur in concatenated error correcting codes.)

4. Building blocks having multiple physical properties, each described by a particular group, can cut down resources by simultaneous execution of multiple transformations. (e.g. electron has position, spin, energy level, etc.)

5. Direct physical implementation of complicated group operations can also reduce depth of computation. Can we construct physical building blocks with large group properties?

. . . contd.

Some Directions (contd.)

6. $(Z_2)^d$ does not provide the minimal set of building blocks for $d > 1$; it contains 2^d points compared to $(d+1)$ points of a simplex. Simplicial geometry can be more efficient for multi-dimensional information processing.

Some Directions (contd.)

6. $(Z_2)^d$ does not provide the minimal set of building blocks for $d > 1$; it contains 2^d points compared to $(d+1)$ points of a simplex. Simplicial geometry can be more efficient for multi-dimensional information processing.

7. Counting the number of available states (i.e. entropy or information) can be carried out for arbitrary configurations of building blocks. But quantification of correlations amongst the building blocks becomes increasingly complicated with increasing dimensionality. Quantum information theory generalises Boltzmann entropy to von Neumann entropy. What would be a more general framework beyond that?