System Science of Virtual Reality:

Toward the Unification of Empirical and Subjective Science

John Ringland

Printed by lulu.com.
Version: 2010-05-14
Electronic Version: www.anandavala.info/SystemSimulation.pdf

ISBN 978-1-4457-6010-0

Preface

The intended audience for this book is:

- empirical[1] scientists who are open to the possibility that physical systems supervene[2] on information processes,
- subjective[3] scientists seeking a rationalist[4] conceptual framework with which to integrate and communicate their knowledge,
- system theorists seeking insights into a mathematical formulation of "general system theory" and a novel mathematical approach toward system analysis and design,
- computer scientists seeking insights into a mathematical model of massively parallel processes and a system-oriented approach to software engineering,
- those seeking to *understand*[5] quantum mechanics and the world-view that arises from it,
- those working toward the unification of the empirical and subjective sciences,
- those seeking rational insight into conscious experience and the nature of reality.

---

[1] Empiricism claims that reliable knowledge can be obtained by observing the *content* of sensory experience.

[2] If 'B' supervenes on 'A' then every aspect of B can be explained by recourse to A.

[3] Subjectivism claims that reliable knowledge can be obtained by observing ones own experiential / cognitive *processes*. The activity of the mind rather than the content.

[4] Rationalism claims that there are sources of knowledge other than sense experience and that these can be rationally verified and formulated into a science. Quantum mechanics is an example of a rationalist science, it is based on mathematical formalisms and postulates entities such as wavefunctions that are not observable but nevertheless real.

[5] It has often been stated that quantum mechanics is impossible to understand in any deep sense, however by shifting to an *information systems* paradigm it can be clearly understood and seen to be necessary rather than paradoxical.

This book aims to provide a concise overview of a novel approach to *system simulation*, and its theoretical implications. This subject touches on many profound topics not traditionally associated with system theory. It also sheds new light on many cutting edge topics.

> "The reception of a new paradigm often necessitates a redefinition of the corresponding science. Some old problems may be relegated to another science or declared entirely "unscientific". Others that were previously non-existent or trivial may, with a new paradigm, become the very archetypes of significant scientific achievement. And as the problems change, so, often, does the standard that distinguishes a real scientific solution from a mere metaphysical speculation, word game, or mathematical play". [1]

The book is primarily about systems and simulation in a very general sense. The primary theme is *linear algebraic simulation*, which is an approach that models and animates a target system using systems of linear equations, which are expressed as matrices and vectors (or their logical equivalents such as graphs or networks).

The fundamental principles of simulation and their correspondence to linear algebraic methods are examined. An initial system modelling methodology is described, which is then gradually refined into a more advanced form. This refinement process naturally results in the derivation of the core mathematical foundations of quantum mechanics, thereby situating quantum mechanics within a computational / system theoretic context. By following this process the reader may develop a deeper understanding of both general system modelling and quantum mechanics.

The process of simulation intrinsically implies the concept of a *virtual reality* in which the observer is both emergent from and embedded within the information process. It also highlights the ramifications of *naïve realism*[6].

The subject of consciousness, in particular subjective experience, is discussed from the perspective of the systems paradigm, which leads to a compelling resolution of the *hard problem of consciousness*[7] [2]. Finally an attempt is made to derive the general form of the *dynamical equations of individual consciousness*.

[6] Naïve realism is an erroneous, wide-spread and mostly sub-conscious cognitive habit of assuming that the contents of experience are actual objects and events and not just cognitive representations. This leads to the belief that the mind provides direct access to reality and that we experience "the world as it is".

[7] The hard problem of consciousness addresses the relationship between subjective experience and objective reality. As Nagel (1974) has put it, there is *something it is like* to be a conscious organism; a subjective aspect to experience.

The underlying concepts are relatively easy to grasp if approached with persistence and genuine scepticism[8]. Hence the discussion will be kept as simple and approachable as possible and will use concrete examples. If there is a choice between approaches, the more accessible approach is used wherever possible. Likewise with references; freely available online resources have been preferred over more difficult to obtain resources wherever possible.

Throughout this book the concepts are developed in a logical sequence and later statements are made within the context of the conceptual framework developed thus far. Hence, those that skip ahead but still interpret things according to the 'old' paradigm, may encounter statements that seem paradoxical. However if they follow the logical sequence of ideas they will see that within the context of the 'new' paradigm these statements represent inevitable logical conclusions.

To get the most from this book an undergraduate level understanding of core aspects of algebra, computer science, system science, quantum mechanics and philosophy may be necessary. As well as experience with subjective contemplation of ones own cognitive processes. However, because we start with elementary principles and develop the concepts systematically this book can be followed with only minimal experience or comprehension in these areas.

Although aspects of this book are quite mathematical the underlying ideas themselves are quite intuitive. Hence the book has been written so that the reader can, if they wish, skim over the mathematics and still follow the discussion. However, for a detailed understanding it is necessary to follow the development of the mathematics closely.

Acknowledgments

This book is the product of over a decade of work, which could not have been accomplished without the support of family and friends. I also owe a great debt to all the thinkers that have contemplated these ideas before me; the giants upon whose shoulders I stand, as Newton put it. Many thanks to Chuck Keiser for proof-reading the manuscript. I would like to personally thank Allen and Minako Ringland, Glistening Deepwater and Sally Dawson for their support and feedback. Finally, a warm thank you to the many people who taught me valuable lessons over the years.

[8] 'Scepticism' involves approaching a subject without prejudice, enquiring into it via all available means, balancing all possible conclusions within ones mind, only believing something once there is compelling evidence and being prepared to adjust ones beliefs and work through the ramification of that once new evidence comes to light.

Contents

Chapter 1

Fundamental Principles

In this chapter we first discuss the fundamental principles of simulation then use a simple example to illustrate how these principles map to the linear algebraic concepts of matrices and vectors. This chapter provides some initial insights into the nature of virtual reality and general systems, as well as some motivation and orientation prior to a deeper exploration of linear algebraic methods for system simulation.

1.1 Simulation

The goal of simulation is to create a simulacrum of a system. To do this we must represent the system in some manner, then devise an algorithm that manipulates that representation so that it changes in a way that resembles the behaviour of the system. In general, to do this we require a *simulator* that uses a *system model* to produce a *simulation* of the system. Both the product of the process and the overall process itself are referred to using the term 'simulation', however it will be clear from the context whether the term is used as a noun or a verb.

The term simulation is often used to refer to cases where we create a computational model of a physical system in order to study its dynamical behaviour, however the process is far more general. All forms of mathematical modelling can be conceived of as a form of simulation and so too can all forms of software engineering.

A simulation can be used to *analyse* systems and make predictions about their behaviour. This underlies the effectiveness of mathematical science, which can model and simulate a wide variety of physical systems. In this case the simulator is traditionally a scientifically trained human mind, the model is a set of mathematical equations as well as theoretical propositions and the simulation is the calculations which produce answers to questions about physical systems.

A simulation can also be used to *create* virtual systems. This underlies the utility of computer software, which can model and simulate systems such as documents, forums and telephone exchanges to the extent that they are functionally equivalent to, or better than, the physical systems upon which they were originally modelled. In this case the simulator is a computational process, the system model is the program as well as data and the simulation is the running application.

The general approach described in this book applies to all forms of simulation and can therefore shed light on aspects of mathematical modelling as well as software engineering.

However, first we introduce the basic principles of systems and the corresponding basic requirements that any simulation procedure must meet in order to accurately simulate them.

1.1.1 Target System

The system that is to be imitated is referred to as the *target system*. Any system may be imitated, whether physical, symbolic, currently existing or imagined.

Systems are composed of sub-systems and integrate to form super-systems. This results in a system hierarchy where each level is defined in terms of the complexity of the systems[1]. An *atomic system* is a system that has no sub-systems and a *complex system* is a system that has sub-systems. An example hierarchy is; particles, atoms, molecules, cells, organisms, civilisations and so on. Or perhaps bits, bytes, codes, programs, operating systems, the internet and so on.

In general the system hierarchy is not rigidly defined or static because systems develop along lines of communication, thus they dynamically integrate and disintegrate with the changing interaction patterns. Therefore, a system may be a sub-system within multiple super-systems. For instance, a person may be a sub-system within a family, a corporation and a nation, whilst the family and corporation are also sub-systems within the nation. Hence there are no rigidly defined system boundaries and the hierarchy is complex and inter-penetrating.

The system model needs to accommodate these basic principles of systems in some manner. For any given target system it may not be practical to model

[1] The complexity of a system relates to the number of sub-systems and the number of inter-connections between sub-systems.

the complete system hierarchy but all relevant details should be represented.

1.1.2 System State

The functional attributes of the target system can be abstracted as its *state*. For instance, the volume of fluid in a reservoir is a functional attribute of a reservoir. A complex system may have many attributes and not all attributes may be functionally relevant. The degree to which the system model captures the functional attributes of a target system is the degree of *sufficiency* of the model.

In general there are two ways to represent a system state within a system model. Either using symbolic variables such as numerical values or as a coordinate within a *state space*, which is the space of all possible states of the system.

The state of a system is an abstraction of a system's attributes because both the target system and the simulated system may exhibit the same state even though the target system's attribute may be a volume of fluid whilst the simulated system's attribute may be a numerical value or a coordinate in a state space. These attributes are quite different however the state is logically equivalent.

1.1.3 Connectivity

In any target system the various sub-systems exhibit a degree of *connectivity*. This can be thought of as lines of communication. Whether or not two sub-systems are communicating in a given moment is not relevant in terms of their connectivity; what is important is their *potential* to communicate. For instance, consider a group of people in a room where each person can potentially communicate with each other; this system exhibits point-to-point connectivity. These connections between systems may be direct or mediated by chains of intervening sub-systems and the network of connectivity may dynamically change. The system model needs to represent the connectivity between sub-systems within the target system.

1.1.4 Interaction

When systems communicate in a given moment and respond to the information that is passed between them they can be said to *interact*. These interactions may occur at any level of the system hierarchy, hence they are not just *between* systems but also *within* systems when they occur between sub-systems. Hence the behaviour of a system arises from the interactions between its sub-systems, whilst at the same time that system is interacting with other systems and thereby manifesting the behaviour of super-systems. In this manner, all events at all levels of the system hierarchy are mediated by system interactions. The simulator needs to implement these interactions in order to produce functionally equivalent events within the simulation.

1.1.5 State Transition

When systems interact they undergo a *state transition* whereby their state changes in response to the interaction. Examples of this are particles changing momentum due to a collision or a corporation changing its marketing strategy due to discussions between members of its marketing department. All changes in state are the effect of a system interaction, either between systems or between sub-systems within a system.

The simulator can implement a state transition either as the change of a state variable or as a transformation within a state space that maps the initial coordinate to some other coordinate.

1.1.6 System Evolution

The changing system states result in a process of *system evolution*. This process arises from the coordinated interactions of many systems, thus exhibiting global coherence, i.e. at each level of the system hierarchy there arises collective entities that are not just masses of individuals but which act as a single coordinated collective. This can be simulated either as patterns of changing state variables or as trajectories within a state space.

1.1.7 Functional Equivalence

In the limit of complete sufficiency there is an exact functional equivalence between the target system and the simulated system (virtual system), such that the two are indistinguishable. For example, telephone exchanges were initially implemented as electro-mechanical switching devices however now the functional attributes are entirely simulated and implemented in software. The resulting application is not only functionally equivalent to the original but due to the flexibility of computational processes it is greatly enhanced.

Another example is the progress of mathematical science, where the models have increased in sufficiency to the point where many aspects of the physical universe can be calculated *ab initio* (from first principles) without resorting to experimental measurements or ad hoc hypotheses. In this sense a fully sufficient science would have a core set of fundamental axioms and be able to answer all questions using ab initio calculations and deductions.[2]

1.1.8 Virtual Systems and Virtual Reality

The simulated systems can be thought of as *virtual systems* and the context in which they perceive[3] each other and themselves can be thought of as a *virtual reality*. This does not just refer to contemporary examples of VR where the observer exists in physical space whilst their senses are stimulated so they experience a virtual world or experiential context. Rather it refers to the more general scenario of an abstract mathematical or computational process that causes abstract systems to emerge from and interact within an abstract information space. Some examples of virtual systems are particles within a particle simulation, conceptual entities within a mathematical model, documents within a word processing software application or characters within a computer game.

In regards to mathematical science the virtual systems are the conceptual entities that are proposed by the theories. For example, in physics there are quarks, gluons, strings, quantum foam, wavefunctions and so on. The virtual reality is the image of the physical universe that is proposed by the scientific theories. The degree to which the virtual systems correspond to physical systems depends on the sufficiency of the simulation process.

In regards to computation the virtual systems are computational entities such as user interfaces, emails, desktops, file systems, network sockets, viruses and so on. Computational virtual reality is also known as cyberspace.

The concept of a simulation can also be extended to include narratives such as novels or movies. For instance, in the case of a novel the simulator is the imagination of the reader, the system model is the text, the simulation is the story as it unfolds in the reader's mind, the virtual systems are the objects, people and places within the story and the virtual reality is the world in which the story takes place. However the methods presented here are primarily suited to mathematical and computational simulation and can only provide an analogy for such narrative simulations.

---

[2] The current effort within physics to find a *theory of everything* (TOE), if successful would produce a sufficient science encompassing the field of physics, however given its current approach it would not encompass the whole of science and as such would be sufficient only within a limited, albeit fundamental domain. There would still remain major outstanding issues such as the *hard problem of consciousness* [2]. For more on the hard problem see sec. 5.3

[3] Here the term 'perceive' refers to a system's reception of input signals and their transformation into internal signals. The more common meaning also implies an experiential aspect, which will be discussed later in relation to consciousness (sec. 5.3).

1.2 Linear Algebraic Simulation

Having introduced the fundamental principles of systems we now illustrate the manner in which linear algebra can be used to model each of these principles.

To facilitate this illustration we will consider an idealised system that is stripped of many complicating factors. We will develop a very basic linear algebraic model of this system and then run the simulation. More advanced methods will be introduced later that can simulate more complex systems, however this initial approach illustrates the basic principles of linear algebraic simulation.

Consider a target system consisting of four water tanks t_0, t_1, t_2 and t_3 connected by pipes. In each iteration a proportion of the water in a tank flows through to other tanks (fig. 1.1).

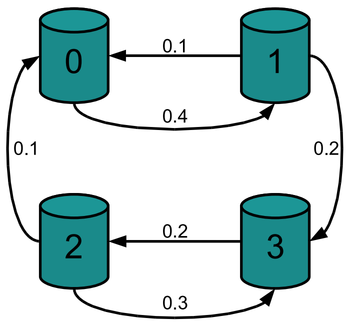

Figure 1.1: *Four water tanks connected by a network of pipes.*

There is no system hierarchy of sub / super systems that needs to be modelled here, thus we will address this issue later (sec. 2.5.1).

1.2.1 State → State Vector

There are two approaches to representing system state, using variables or state spaces, however here we will only consider the use of variables (state spaces will be discussed in chapter 2).

There are four systems, each with a single functional attribute so we can use a vector V with four elements v_0, v_1, v_2, v_3 where v_i represents the volume of fluid in tank t_i. Thus

$$V = \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} \tag{1.1}$$

1.2.2 Connectivity → Causal Matrix

In our example target system there is a particular network of connections between tanks (shown in fig. 1.1), however in general each tank can potentially be connected to each other tank. Hence the system model must be capable of representing any possible connective network as well as specifying a particular connective network.

If we think of the situation in terms of a weighted digraph (directed graph) where the tanks are nodes and the connections are edges then we get the digraph shown in fig. 1.2.

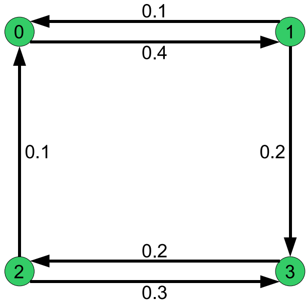

Figure 1.2: *Weighted digraph of the connectivity between tanks.*

A weighted digraph can be represented using a matrix M where each connection between nodes is represented by an element in the matrix:

$$m_{ij} = \begin{cases} w & \text{if an edge connects from node } j \text{ to node } i, \\ 0 & \text{otherwise.} \end{cases}$$

Where m_{ij} refers to the element in the i^{th} row and the j^{th} column of the matrix and w is the *weight* of the connecting edge.

The resulting causal matrix represents the connectivity *between* tanks so when used in a simulation it would model the flow of water between tanks, however we also need to model the water that remains in each tank and does not flow.

Hence the causal matrix also includes *self-interaction* terms, which lie along the diagonal of the causal matrix. This models the flow between a tank and itself, i.e. the water that remains in that tank.

$$M = \begin{pmatrix} 0.6 & 0.1 & 0.1 & 0 \\ 0.4 & 0.7 & 0 & 0 \\ 0 & 0 & 0.6 & 0.2 \\ 0 & 0.2 & 0.3 & 0.8 \end{pmatrix} \tag{1.2}$$

Each column of the causal matrix contains information about a system's *outputs*. For instance, by looking down the first column we can see which tanks t_0 outputs to and what proportion of its water flows to that tank. We see $m_{00} = 0.6$ hence 60% of its water remains in that tank, $m_{10} = 0.4$ hence 40% of its water flows to t_1.

Notice that each column is *normalised* (sums to one). If a column summed to less than one water would be lost from the system and if it summed to more than one a tank could output more water than it contained thereby creating water within the system. This damping and amplification can be useful in some simulations but it is not appropriate in this simplistic example where water is a conserved quantity.

Each row of the causal matrix contains information about a system's *inputs*. For instance, by looking across the bottom row we can see which tanks t_3 receives inputs from and what proportion of the source tank's water flows into t_3. We see $m_{31} = 0.2$ hence 20% of t_1's water flows into t_3, $m_{32} = 0.3$ hence 30% of t_2's water flows to t_3 and $m_{33} = 0.8$ hence 80% of t_3's water remains in t_3.

Thus we see that there is a logical equivalence between the matrix representation, the weighted digraph representation and the connectivity of the target system.

1.2.3 Interaction → Matrix / Vector Multiplication

System interactions are mediated by the transmission of information through the connective network. This can be represented using matrix / vector multiplication ($M \cdot V$). To visualise how this works imagine that the state vector is rotated into a row vector $\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \end{pmatrix}$ and each element of the state vector is paired with the elements of a matrix column.

$$\begin{array}{c} \begin{pmatrix} v_0 & v_1 & v_2 & v_3 \end{pmatrix} \\ \begin{array}{cccc} \downarrow & \downarrow & \downarrow & \downarrow \end{array} \\ \begin{pmatrix} 0.6{\cdot}v_0 & 0.1{\cdot}v_1 & 0.1{\cdot}v_2 & 0{\cdot}v_3 \\ 0.4{\cdot}v_0 & 0.7{\cdot}v_1 & 0{\cdot}v_2 & 0{\cdot}v_3 \\ 0{\cdot}v_0 & 0{\cdot}v_1 & 0.6{\cdot}v_2 & 0.2{\cdot}v_3 \\ 0{\cdot}v_0 & 0.2{\cdot}v_1 & 0.3{\cdot}v_2 & 0.8{\cdot}v_3 \end{pmatrix} \begin{array}{c} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{array} \begin{pmatrix} v'_0 \\ v'_1 \\ v'_2 \\ v'_3 \end{pmatrix} \end{array} \tag{1.3}$$

We can visualise the value contained in a vector element flowing down a column, where each matrix element in the column deflects a proportion of the value off to the right where it accumulates in a new vector. In this way all of the initial vector elements flow through the causal matrix to be distributed into the new vector. Thus the causal matrix defines the causal relation between successive system states, hence it is called a *causal matrix.*

In the context of the model each pair calculates the proportion of water that flows through a particular pipe. Then each row (system input) is summed, thus calculating the total amount of water that flows into a particular tank. Thus:

$$\begin{aligned} M{\cdot}V_t &= \begin{pmatrix} 0.6 & 0.1 & 0.1 & 0 \\ 0.4 & 0.7 & 0 & 0 \\ 0 & 0 & 0.6 & 0.2 \\ 0 & 0.2 & 0.3 & 0.8 \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} \\ &= \begin{pmatrix} 0.6{\cdot}v_0 + 0.1{\cdot}v_1 + 0.1{\cdot}v_2 + 0{\cdot}v_3 \\ 0.4{\cdot}v_0 + 0.7{\cdot}v_1 + 0{\cdot}v_2 + 0{\cdot}v_3 \\ 0{\cdot}v_0 + 0{\cdot}v_1 + 0.6{\cdot}v_2 + 0.2{\cdot}v_3 \\ 0{\cdot}v_0 + 0.2{\cdot}v_1 + 0.3{\cdot}v_2 + 0.8{\cdot}v_3 \end{pmatrix} \end{aligned} \tag{1.4}$$

1.2.4 State Transition → New State Vector

The vector resulting from the iteration becomes the state vector for the next moment of time within the simulation.

$$\begin{pmatrix} 0.6{\cdot}v_0 + 0.1{\cdot}v_1 + 0.1{\cdot}v_2 + 0{\cdot}v_3 \\ 0.4{\cdot}v_0 + 0.7{\cdot}v_1 + 0{\cdot}v_2 + 0{\cdot}v_3 \\ 0{\cdot}v_0 + 0{\cdot}v_1 + 0.6{\cdot}v_2 + 0.2{\cdot}v_3 \\ 0{\cdot}v_0 + 0.2{\cdot}v_1 + 0.3{\cdot}v_2 + 0.8{\cdot}v_3 \end{pmatrix} = \begin{pmatrix} v'_0 \\ v'_1 \\ v'_2 \\ v'_3 \end{pmatrix} = V_{t+1} \tag{1.5}$$

Hence the information representing the previous state of the system has flowed through a *causal framework* and changed into information representing a new state of the system. Thus $V_{t+1} = M{\cdot}V_t$ and as this equation is iterated the system undergoes a sequence of state transitions.

Exercise: To get a feel for the process of matrix / vector multiplication and what it means for a system model, consider the following two models:

$$\begin{pmatrix} 0.3 & 0.1 \\ 0.7 & 0.9 \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 4 \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} 0.6 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.4 \\ 0.0 & 0.3 & 0.3 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 3 \\ 5 \end{pmatrix}$$

For each model, draw the corresponding directed graph or system network, then calculate the states of the system for the next few iterations. Observe how the information flows from the state vector, through the causal matrix and back into a new state vector that represents the next state of the system.

1.2.5 System Evolution → Iteration

Lets explore the evolution of the system of tanks by defining an initial state for each tank then iterating the equation. Let the state variables represent the number of litres of water in a tank and the initial state of the system is:

$$V_0 = \begin{pmatrix} 40 \\ 30 \\ 20 \\ 10 \end{pmatrix} \tag{1.6}$$

We then iterate $M{\cdot}V$ a number of times and graph the successive state vectors, showing the time evolution of the overall system (fig. 1.3).

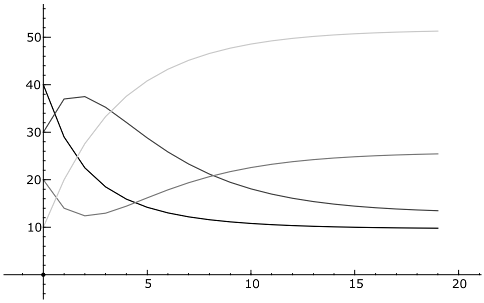

Figure 1.3: Evolution of a system of water tanks over time.

From this simulation run we can observe the evolution of the system over time. However the general behaviour of the system could have been inferred from the weighted digraph (fig. 1.2) by noting that tanks t_0 and t_1 form a feedback loop f_{01} where t_1 would tend to accumulate the available water. Similarly t_2 and t_3 form a feedback loop f_{23} where t_3 would tend to accumulate. We can also note that f_{01} and f_{23} also form a feedback loop where f_{23} would tend to accumulate.

We can analyse the behaviour of the systems further by noting that each successive state vector is produced by the application of the causal matrix to the current state vector.

$$\begin{aligned} V_1 &= M{\cdot}V_0 \\ V_2 &= M{\cdot}V_1 \\ &\vdots \\ V_t &= M{\cdot}V_{t-1} \end{aligned} \tag{1.7}$$

Each iteration multiplies the previous result by the causal matrix M. For example, by substitution we see that $V_2 = M{\cdot}V_1 = M{\cdot}M{\cdot}V_0 = M^2{\cdot}V_0$, and in general

$$V_t = M^t{\cdot}V_0 \tag{1.8}$$

In this manner we can calculate the state of the system at any future moment without having to iterate through each intervening moment.

For instance, the system state after 1000 and 2000 iterations is:

$$M^{1000}{\cdot}V = \begin{pmatrix} 9.67741935483919 \\ 12.9032258064522 \\ 25.8064516129045 \\ 51.6129032258090 \end{pmatrix}$$

$$\text{and} \tag{1.9}$$

$$M^{2000}{\cdot}V = \begin{pmatrix} 9.67741935483967 \\ 12.9032258064529 \\ 25.8064516129058 \\ 51.6129032258116 \end{pmatrix}$$

From this we see that the system tends toward a steady state equilibrium where the feedback loop f_{23} between t_2 and t_3 has accumulated most of the available water. Although tank t_0 initially had the most water, the network of pipes gives rise to a system evolution that results in t_0 eventually having the least amount of water.

In the above example there is still a minor change between steps 1000 and 2000. The condition for a true steady state is $V_{t+1} = V_t$ thus $M^n = I_{4\times4}$ where $I_{4\times4}$ is the 4×4 identity matrix $I \cdot V = V$.

Note: Many calculations performed throughout this book were carried out using Sage [3], an open-source computer algebra system (CAS), and Sage code embedded in the LaTeX code of this document.

1.2.6 Non-Linear Time-Variant Systems

A linear algebraic equation such as the one above is a linear time-invariant system. It is linear because the output is always a simple function of the input ($V_o = M \cdot V_i$) and it is time-invariant because the same input signal always produces the same output signal regardless of when it is applied.

However it can still simulate non-linear and time-variant systems. Whilst the overall $M \cdot V$ iterative equation is entirely linear and time-invariant, within it is simulated a virtual system. When the simulated inputs and outputs are considered the virtual system can be non-linear and time-variant. To illustrate this we will examine a particular example.

The following model implements a simple echo chamber or resonant cavity. This system model is not meant to be a realistic model of a resonant cavity, however it is close enough to be non-linear and time-variant. Any input signal enters the echo chamber, where any subsequent input is also added to the echo. The echoes in the echo chamber leak out slightly so that they gradually die down. Any signal that leaks out goes to the output channel.

The system model is:

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0.998 & 0 \\ 0 & 0.002 & 0 \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ v_0 + 0.998 v_1 \\ 0.002 v_1 \end{pmatrix}$$

The system has three sub-systems; an input channel (S_0), an echo chamber (S_1) and an output channel (S_2).

Aside: The input channel can be discerned by the row of zeroes, which means that system S_0 does not receive any input from any of the other sub-systems, thus any input it receives must come from some external source. Similarly, the output channel can be discerned by the column of zeroes, which means that system S_2 does not transmit any information to any of the other sub-systems, thus any output it produces must go to some external source. Thus the system model represents an *open system*. Open system matrices can be placed along the diagonal of a larger causal matrix and inter-connected by overlapping certain rows and columns, which connects the channels. In this way larger systems can be constructed from numerous open systems. In general a complete system model with a single causal matrix and state vector is a *closed system* because the iterative process is self-contained. The state vector represents the complete information space and the causal matrix represents the complete causal framework for an entire simulation universe.

If the system model as a whole is conceptualised as a function (using Sage [3] notation):

```
def simulator(V): # input V is a vector
    global M      # use a pre-defined matrix
    return M*V    # outputs a vector
```

This function is linear because $V_o = M \cdot V_i$ and it is time-invariant because if $V_o = \text{simulator}(V_i)$ then an input of V_i always produces an output of V_o. However if we consider the virtual system as a function:

```
def echo(x):        # input x is a scalar value
    global V, M     # use a pre-defined vector and matrix
    V[0] = x        # sets the input channel to x
    V = M*V         # performs one iteration of the model
    return V[2]     # returns the output channel (scalar)
```

This function is non-linear and time-variant. If $y = \text{echo}(x)$ then y is not a simple function of x because the output depends on the state of the echo chamber, which changes, hence it is also time-variant. A given input value x_i will produce different output values at different moments.

To observe this behaviour we graph the echo() function over 6000 iterations (fig. 1.4).

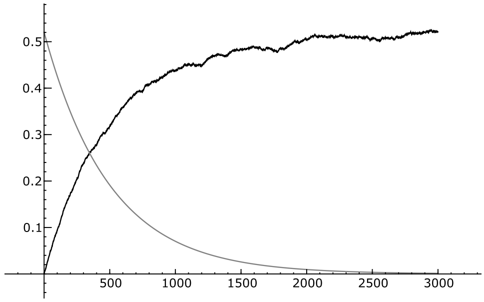

Figure 1.4: Random input echo chamber.

We drive the echo chamber with a random input signal $0 \leq x < 1$ until it is resonating strongly (black line) then we set $x = 0$ (null input signal) and observe the echoes die down to silence (grey line). Note: the grey line follows on from the black line however the two phases are overlayed in the graph to conserve space.

The black line reaches a saturation point, where on average the same amount of signal is leaking out as is being supplied. Furthermore, although the input is random between zero and one, the output remains relatively stable at 0.5. This illustrates a common use for an optical cavity, e.g. to stabilise a fluctuating laser beam.

From this example we see that although the simulator is linear and time-invariant it is not limited to only simulating linear time-invariant systems. Thus a broad range of systems can be simulated using linear algebraic methods.

1.2.7 Virtual Systems and Virtual Reality

Having now developed an initial mathematical model that captures the fundamental principles of a system we can further clarify the concepts of virtual systems and virtual reality. Whilst we describe these within the context of the model just defined, the general principles apply to all modelling frameworks.

Within the iterative $V_t = M \cdot V_{t-1}$ system model we see that a virtual system is an emergent property of the functioning of a simulator as it acts upon a system model to produce a simulation.

There are four aspects of a virtual system, which are distributed throughout the system model.

1. It has a state that is represented by information within the state vector.

2. It has *interfaces*:

    - an input interface that is represented by a row of the causal matrix,
    - an output interface that is represented by a column of the causal matrix,
    - a self-interaction channel where its input and output interfaces intersect,
    - input channels where its input interface intersects another system's output interface,
    - output channels where its output interface intersects another system's input interface.

3. It has an *experiential process*[4], which is the operation of the simulator (*animating process*) as it integrates the input information, computes a new system state and thereby animates a virtual system.

4. It has an observable form or *appearance*, which arises when its state information flows through an input interface and is perceived by a system's experiential process.

By taking these four aspects of a virtual system into account and considering the perspective of a virtual system, this results in four aspects of a virtual reality.

1. A virtual system can only perceive the information that flows through its input interface and is processed by its experiential process. This information only informs the system of the appearances of itself and other systems. Hence the other aspects such as state, interfaces and animating process are imperceptible.

2. Thus from a system's perspective it appears as if it is an individual entity that simply exists within a space filled with interacting entities, all of which are known by their appearances.

3. The interactions between these entities are tangible[5] and for all intents and purposes they seem 'physical' because all systems are emergent from and embedded within the same information space.

4. The systems respond to the perceived appearances as if they were all that exists because this is all that they can experience. Thus a virtual reality is a construct of the appearances of systems and the virtual systems experience that virtual reality as their *physical universe*.

These four aspects of virtual systems and virtual reality are not dependent on the type of simulation methodology used but are fundamental principles of simulation itself. From this we see that from a virtual perspective most of reality is imperceptible and the observable part is a *film* of appearances that forms within the input interfaces of the virtual systems themselves.

---

[4] To call the animating process operating 'within' a virtual system an 'experiential' process may seem unusual to some however as we explain the systems paradigm further this will become clearer. For now let this phrase loosely describe the fact that a virtual system receives input, responds to it and is affected by it. Once we have examined quantum systems and then come to discuss consciousness we are in a position to say more about the subjective aspect of what we are refering to here as the experiential process.

[5] The interactions are 'tangible' because an input signal has a definite impact on the state of a system.

1.2.8 System Matrix Notation

It should now be apparent that linear algebra provides a very compact and elegant method for system simulation. The simple examples given above illustrate that the fundamental principles of systems and system simulation can be directly mapped to the structure and interaction of matrices and vectors as well as graphs and networks. The simplicity and mathematical foundations of this approach lend it to further development, of both its modelling capabilities and the range of analyses that can be performed on the system models. In the following chapters we will explore more complex variations upon this theme and use these methods to model more complex systems.

However not all variations are strictly *linear* or strictly *algebraic*, as we will see in sec. 4.1. Due to these variations this overall approach to system simulation is given the more general name *System Matrix Notation* (SMN) because it is a mathematical notation and methodology that uses matrix-oriented methods to model and simulate general systems.

Chapter 2

Probabilistic Simulation (PSMN)

In this chapter we explore a slightly more abstract application of linear algebraic methods. These methods are still limited but they illustrate some deeper aspects of the overall approach.

The method illustrated earlier used real valued variables to represent system state, however here we encounter the use of discrete state spaces, and real valued probability distributions over those state spaces. This approach is called probabilistic system matrix notation (PSMN).

PSMN is a useful stepping stone toward a more complete methodology because it is simple and intuitive yet it exhibits many of the core principles of the later more abstract approaches. Thus we can use it to map out the overall structure of a state space oriented SMN simulator. By identifying the limitations of the PSMN approach and overcoming those limitations we naturally derive the foundations of quantum mechanics.

First we must clarify some of the basic principles of the linear algebraic approach. In this chapter we illustrate these basic principles in the context of real valued vector spaces however later they will be extended to more complex spaces.

2.1 Information

First some fundamental concepts regarding *information* should be introduced. There are several definitions of information that are useful in different contexts, however here we take a very general approach and define information as *discernible difference*. This requires an *information medium* which is able to exhibit two or more different states and an *information process* that is able to discern those states as being different. Therefore, information is an emergent property of the interaction between an information process and an information medium.

An information process not only discerns but also modifies the different states of an information medium. The set of all possible states that can be exhibited by an information medium is called an *information space*. This is the field of discernible difference within which the information process operates.

For example, if an information space consists of the set of integers, or all real numbers with absolute values less than one, then any mathematical function on that space can only operate within those constraints. If an information space consists of 64bit binary words then a computer program can only perform 64bit precision calculations unless it implements its own specific algorithms to effectively enlarge the information space by using multiple words.

2.2 Rings

An information space is related to what mathematicians call a *ring*. To situate rings within the context of linear algebraic simulation consider that vectors and matrices can be thought of as structured containers for variables. A variable is a symbolic entity within a computation that has both an identity (e.g. the variable x or the vector element v_n) and a set of possible values (e.g. $x \in \mathbb{R}$ which means that x exhibits values from the set of all real numbers).

A ring defines the possible values that a variable may exhibit and therefore defines the information space, which places fundamental constraints on the operation of all information processes within that space.

For now the only rings that we will consider are $\mathbb{B}$ binary (0, 1) and $\mathbb{R}$ the set of all real numbers. We will explore some of the properties of information processes within this context and then later we will encounter complex rings.

2.3 Vector Spaces

A vector can be thought of as an ordered list of variables which can exhibit values defined by a ring. For example, consider the vector $v = \begin{pmatrix} v_0 & v_1 \end{pmatrix}$ with $v_i \in \mathbb{R}$. The set of all possible states of a vector is a *vector space*. Thus a particular instance of the vector v is a coordinate that identifies a point within a 2D Euclidean vector space and the set of all possible instances of v spans an entire 2D Euclidean vector space.

The number of elements in a vector (its degree) determines the number of dimensions of the corresponding vector space. Each component of the vector represents a magnitude along an orthogonal (perpendicular) axis of a vector space. For instance, v can be expressed as a linear combination of two orthogonal vectors $v = c_0 \begin{pmatrix} 1 & 0 \end{pmatrix} + c_1 \begin{pmatrix} 0 & 1 \end{pmatrix}$ where coefficients $c_i \in \mathbb{R}$.

A set of orthogonal vectors that span a vector space is called a basis set and the individual vectors are called basis vectors. Any vector within the vector space can be parametrised as a linear combination of basis vectors. Thus $V = \sum_{i=1}^{n} c_i \cdot v_i = c_1 \cdot v_1 + \ldots + c_n \cdot v_n$ where the c_i are real valued coefficients, the v_i are basis vectors and n is the dimension of the vector space as well as the degree of the vectors.

Any basis set can be used, for instance the axes of the Euclidean space can be rotated, thus the values of the coefficients will change but the point itself will remain unchanged.

A more abstract (curvilinear) parametrisation of the two dimensional space is (r, θ), which uses distance from the origin and angle from the positive x axis to specify the coordinate of the point (fig. 2.1).

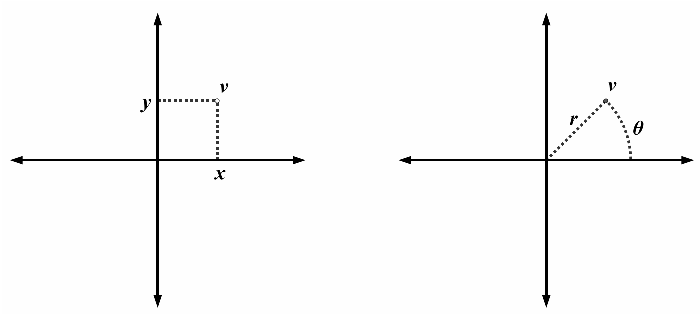

Figure 2.1: Two Coordinate Schemes for a 2D Euclidean Plane.

It doesn't matter which coordinate scheme is used, so long as there is enough information to uniquely specify the point. However in particular cases it is often much simpler to use one coordinate scheme over another. In this book we use a Euclidean parametrisation.

2.4 State Vectors

When a vector is used to represent the state of a system it is called a *state vector*. The particular type of state vector that we will examine in this chapter contains a single element for each possible state that a system may occupy. These vector elements are real valued variables in the range $0 \leq v_i \leq 1$, which represent the probability that the system is currently in that state. The state vector represents a probability distribution over all possible states. It must be *normalised*, i.e. the sum over all probabilities must be equal to one because it is certain that the system exhibits some state, but it is not necessarily certain which state.

To illustrate this with a concrete example, consider a binary system that can occupy only two possible states that we will call s_0 and s_1 (the methods illustrated here can be directly applied to systems with more states). The state vector will then have a degree of 2 (two elements) $v = \begin{pmatrix} p(0) \\ p(1) \end{pmatrix}$ where $p(0)$ is the probability that the system is in state s_0 and $p(1)$ is the probability that the system is in state s_1. This is refered to as a *qubit*. If $v = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ then $p(0) = 1$ so the system is definitely in state s_0 and if $v = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ then $p(1) = 1$ so the system is definitely in state s_1.

If we restrict the probabilities to only integer values (0 or 1) then we retain a strictly classical representation where a system must be entirely in one state OR the other. In this case we refer to the state vector as *pseudo-classical* because it is a probability distribution that represents a classical state. By using real valued probabilities we move one step closer to quantum mechanics because a system can be in a mixed state but there are still major differences between this probabilistic approach and quantum mechanics. However, what we learn here provides a stepping stone towards the quantum case.

2.5 Direct Products

So far we have defined the state of a single isolated system, but in order to model system interactions we need to consider their joint probabilities. Consider two binary systems $A = \begin{pmatrix} p_a(0) \\ p_a(1) \end{pmatrix}$ and $B = \begin{pmatrix} p_b(0) \\ p_b(1) \end{pmatrix}$. The direct product of these two vectors gives us a vector that contains all permutations of elements from each vector.

$$A \otimes B = \begin{pmatrix} p_a(0) \\ p_a(1) \end{pmatrix} \otimes \begin{pmatrix} p_b(0) \\ p_b(1) \end{pmatrix} = \begin{pmatrix} p_a(0)p_b(0) \\ p_a(0)p_b(1) \\ p_a(1)p_b(0) \\ p_a(1)p_b(1) \end{pmatrix} = \begin{pmatrix} p_{ab}(00) \\ p_{ab}(01) \\ p_{ab}(10) \\ p_{ab}(11) \end{pmatrix} \tag{2.1}$$

In general if the two vectors have degree N_a and N_b then the resulting vector will have degree $N_a \cdot N_b$.

2.5.1 Joint Probability / Complex Systems

The resulting state vector represents states of the combined system AB. Thus the model is no longer of two isolated systems A and B but of the single system AB that exhibits four states s_{00}, s_{01}, s_{10} and s_{11}. This addresses the issue of representing the system hierarchy that was mentioned in regards to the target system (sec. 1.1.1).

This approach also highlights the connection between interaction and complex systems. If the two systems A and B interact then we need to model their joint probability, likewise if they interact they participate as sub-systems of the complex system AB. Thus the fact of interaction and the fact of being sub-systems are different ways of thinking about the same situation. If a group of systems are strongly interacting it is perceived as a single complex system whereas weakly interacting systems are perceived as separate systems.

The process whereby a group of sub-systems is perceived to become a single super-system is called a *meta system transition*. However, as we see here, nothing fundamentally new is created when a super-system seemingly comes into existence; rather the super-system is an emergent property of coordinated sub-system interactions.

Thus a meta system transition is a perceptual process and different systems will perceive different system hierarchies. When systems interact strongly we perceive them as sub-systems within a single super-system (object of perception) and when they interact weakly we perceive them as the space between the objects of perception.

For example, consider the molecules within a rock, if we apply a force to a small number of them by pushing the rock with a finger the impulse force is rapidly communicated amongst the molecules in the form of a shockwave. Due to the strong interactions all of the molecules begin to move in response to the force and we perceive the rock itself to move as an individual entity.

If the molecules were weakly interacting, such as in a gas, the impulse force would be sparsely communicated and the movement would be less organised, i.e. turbulence within the gas. An intermediate example is a blob of fluid-like jelly, where some impulse forces would result in a clearly moving shockwave and collective motion whereas others would result in the breakup of the blob into what appears to be many blobs.

If a system with different interaction properties (e.g. perceptual faculties) were to observe these same events it would perceive the system boundaries differently. It may perceive the rock as a gas or perhaps the gas as a solid object, depending on the way in which it interacted with the objects. For instance, a neutrino can pass through an entire planet with very little chance of interacting, thus neutrinos would perceive the Earth as "empty space" or a very rarified gas.

This illustrates that the objects that a system perceives are not fundamental entities in themselves. Rather they are objects of perception that are derived from the complex interactions of sub-systems, which produce collective behaviour that may be perceived as a single super-system if the observer has the appropriate perceptual faculties.

Hence system hierarchies are not 'hard-wired' and different systems perceive different system hierarchies and even the same system, in different perceptual modes can perceive different system hierarchies.

This also applies to a system's perception and conception of itself! For instance, one may conceive of oneself as a single indivisible entity (a person), or as an organised collective of trillions of cells (organism), or as a member of a civilisation (citizen).

These conceptualisations correspond to system (individual), super-system (collective) and sub-system (part). Each part or collective can perceive itself as an individual entity and each individual can perceive itself as both a part and a collective.

All of these conceptualisations are equally accurate because system dynamics occur at all levels of the system hierarchy thus no system or level has a privileged position. However each system perceives a universe centered around itself and its level of the system hierarchy, thus it comes to conceive of things from that perspective. A systems based approach can integrate the many individual perspectives in any given context, to provide a holistic overview.

2.6 Matrix Operators

Recall that a vector represents a point within a vector space. Thus, when a causal matrix operates on a vector and transforms it into another vector this is logically equivalent to a transformation within the vector space. A particular causal matrix represents a transformation operator, which can be applied to any vector within the vector space and will transform it into another vector within that space.

Now that we have a state vector for the complex system AB we can consider the form of the causal matrix that will operate on it. First we will consider the case where the causal matrix elements m_{ij} are drawn from a binary ring, with values 0 and 1 only, then we will extend this to the case of probabilities $m_{ij} \in \mathbb{R}$ where $0 \leq m_{ij} \leq 1$.

In the binary case we find that the causal matrix defines a deterministic causal framework. For example, lets study the causal matrix that implements a causal framework for a circuit of two logic gates $A' = A$ NAND B and $B' = A$ XOR B (fig. 2.2).

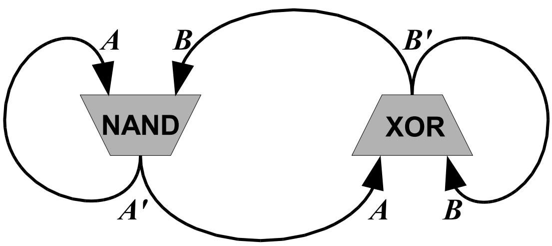

Figure 2.2: A circuit consisting of two logic gates NAND and XOR.

These are two simple binary logic operations between A and B. NAND is NOT AND, which is true when neither A nor B are true. XOR is the exclusive OR, which is true when A or B is true but not both.

The *truth table* for these two logic operations is:

A	B	A'	B'
0	0	1	0
0	1	1	1
1	0	1	1
1	1	0	0

(2.2)

In this truth table the two left columns indicate initial values for A and B and the two right columns indicate the resulting values A' and B' after the application of the logical operations. By reading across the rows, in terms of joint states of AB we see that:

$$
\begin{array}{ccc}
AB & & AB' \\
00 & \rightarrow & 10 \\
01 & \rightarrow & 11 \\
10 & \rightarrow & 11 \\
11 & \rightarrow & 00
\end{array}
\tag{2.3}
$$

From these state transitions we can determine the structure of the required causal matrix. There are two ways of approaching this. For each initial state we look down the corresponding column and place a 1 in the row that corresponds to the resulting state. For instance, in the causal matrix below, consider the transition of state $00 \rightarrow 10$. These are the first and third states so we look down the first column and place a 1 in the third row. Then we perform a similar operation for the other state transitions.

$$
M = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}
\tag{2.4}
$$

Another approach is to consider each input state and the corresponding output state, for example

$$
s_{00} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightarrow \quad s_{10} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}
\tag{2.5}
$$

Then think of the causal matrix as a simple transformation table, so the ouput state s_{10} is placed in the first matrix column, which corresponds to the input state s_{00}. When this is done for each pseudo-classical input we end up with the above causal matrix. Each vector element is like a signal and each matrix column is like a network that distributes the signal along different channels, resulting in a new state vector. A pseudo-classical state fires a full power signal down a particular column of the causal matrix, which distributes this signal into the appropriate output state. This approach is especially useful for visualising matrix / matrix multiplication $M_i \cdot M_j$, which we will encounter later.

When the resulting causal matrix is applied to a state vector it transforms the probabilities in a manner that simulates the behaviour of the two logical operations.

$$M \cdot V = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} p(00) \\ p(01) \\ p(10) \\ p(11) \end{pmatrix} = \begin{pmatrix} p(11) \\ 0 \\ p(00) \\ p(01) + p(10) \end{pmatrix} \tag{2.6}$$

A simple way to test this is to input a pseudo-classical state vector and see what happens to it. For instance, lets consider the case where the initial state is s_{10} so $p(10) = 1$ and the rest are zero.

$$M \cdot V = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{2.7}$$

This implies that $10 \rightarrow 11$ as expected from (2.3). From the bottom row of (2.6) we see that the new $p(11)' = p(01) + p(10)$ thus all of the probability from $p(10) = 1$ has flowed through the causal framework and resulted in $p(11)' = 1$.

We needn't only consider pseudo-classical inputs, the causal matrix operates equally well on general probability distributions. For example, we can find out what happens if the system is in a mixed state, 40% s_{00} and 60% s_{10}.

$$M \cdot V = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0.4 \\ 0 \\ 0.6 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0.4 \\ 0.6 \end{pmatrix} \tag{2.8}$$

Thus the resulting state is 40% s_{10} and 60% s_{11}.

A binary causal matrix such as this is deterministic because each pseudo-classical state is mapped to another pseudo-classical state. However if the matrix elements were probabilities it can be the case that a system that is in a pseudo-classical state may transform into a mixed state. This would result in a probabilistic causal framework where, for instance, being in state s_p may mean that in the next moment the system has a 20% chance of being in state s_q and an 80% chance of being in state s_r.

2.6.1 Non-Algorithmic Processes

One of the benefits of the state space approach can now be explained. Above we saw this approach applied to the simple case where $A' = A$ NAND B and $B' = A$ XOR B, which is a simple algorithmic process. However if we change the causal matrix slightly we can easily produce a simulation of a process that has no algorithmic representation. Therefore, using the state space approach we can simulate all processes, even non-algorithmic processes for which there is no non-state-space representation.

2.7 Dynamical Groups

We saw above that a set of vectors form a vector space and a matrix implements a transformation operation within that space. So let us here consider the set of all matrices in conjunction with the set of all vectors.

Let $\mathbb{M}_{NR}$ be the set of all column normalised, $N \times N$ matrices over the ring R. Let $\mathbb{V}_{NR}$ be the set of all normalised, degree N vectors over the ring R. For simplicity we will consider the case where $N = 2$ and $R = \mathbb{B}$ the binary ring, which represents a deterministic simulation of two classical binary systems. The method illustrated below can be directly applied to more complex simulations.

$$\mathbb{M}_{2\mathbb{B}} = \left\{ \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right\} \tag{2.9}$$

The first matrix always sets the vector to $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ regardless of the initial state.
The second matrix always sets the vector to $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ regardless of the initial state.
The third matrix is the identity matrix which leaves the initial state unchanged.
The final matrix flips the state so that $\begin{pmatrix} 1 \\ 0 \end{pmatrix} \to \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and vice versa.

These matrices can be formed into two subsets ${}^0\mathbb{M}_{2\mathbb{B}}$ and ${}^1\mathbb{M}_{2\mathbb{B}}$. ${}^0\mathbb{M}_{2\mathbb{B}}$ contains the first two operators, which disregard the initial state and ${}^1\mathbb{M}_{2\mathbb{B}}$ contains the last two, which transform the initial state. In the first subset there is one such matrix for each possible state, hence there are N of these matrices. The second subset represents N transformations within the vector space, hence we will consider this subset.

$${}^1\mathbb{M}_{2\mathbb{B}} = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right\} \tag{2.10}$$

For the vectors there are only two possible states.

$$\mathbb{V}_{2\mathbb{B}} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\} \tag{2.11}$$

These sets are particularly simple in this case but the same general principles apply to all finite sets of causal matrices and state vectors.

Now we consider both ${}^{1}\mathbb{M}_{2\mathbb{B}}$ (the set of N transformations) and $\mathbb{V}_{2\mathbb{B}}$ (the set of N states) under the operation of matrix / vector multiplication. This produces a *dynamical group*.

$$\begin{aligned}
\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \cdot \begin{pmatrix}1\\ 0\end{pmatrix} &= \begin{pmatrix}1\\ 0\end{pmatrix}\\
\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \cdot \begin{pmatrix}0\\ 1\end{pmatrix} &= \begin{pmatrix}0\\ 1\end{pmatrix}\\
\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} \cdot \begin{pmatrix}1\\ 0\end{pmatrix} &= \begin{pmatrix}0\\ 1\end{pmatrix}\\
\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} \cdot \begin{pmatrix}0\\ 1\end{pmatrix} &= \begin{pmatrix}1\\ 0\end{pmatrix}
\end{aligned} \tag{2.12}$$

This is a listing of every possible $M \cdot V$ combination. We see here that for every possible pair of state vectors V_j and V_k within this group there exists exactly one matrix operator that transforms $V_j \rightarrow V_k$ and exactly one matrix operator that performs the inverse transformation from $V_k \rightarrow V_j$. This means that every state is accessible from every state and for every transformation there is an inverse.

There are N states and N transformations so for any given state V_j there is the identity matrix that maps V_j to itself and then there are $N - 1$ matrices that map V_j to each of the other possible states.

This is related to $V_t = M^t \cdot V_0$ (1.8), which implies that for any multi-step process from one state to another there exists a logically equivalent single step process. This is remeniscent of a *hyperspace jump* in science fiction. Rather than travel from one state to another by executing all of the intervening steps, it is possible to jump directly to the final state in a single step.

If we select a particular state vector V_j and a particular matrix M_q (which is not the identity matrix) then by repeated application of that matrix we find that $V_j \rightarrow V_k \rightarrow \ldots \rightarrow V_x \rightarrow V_j$ which represents a complete cycle. Hence the repeated application of any causal matrix produces a cyclic system evolution. The identity matrix produces a trivial cycle of one iteration. The repeated application of the matrix M_q for an n step cycle is logically equivalent to the application of a single identity matrix, thus $M_q^n = I$. This cyclic behaviour is trivial in the simple $(N = 2, R = \mathbb{B})$ example because there are only two possible states, however by exploring more complex $(N, R = \mathbb{B})$ scenarios this cyclic behaviour becomes more apparent.

From this analysis we see that the state space approach can simulate the state of a discrete system and its state transitions. From this field of possibility we can select particular matrices and particular initial states that produce a simulation. However the PSMN approach cannot simulate every conceivable discrete system as we will see in sec. 2.8. However once the limitations of PSMN are overcome the resulting state space approach can provide a complete modelling methodology with which to simulate discrete systems.

This discrete approach can also be extended to continuous systems however that is beyond the scope of this book.

2.8 Limitations of PSMN

As a mathematical model PSMN is an improvement on the approach that we used to model the water tanks. However it is generally applied to systems with discrete states, because a single real valued state variable has an infinite number of possible values hence its state vector would have an infinite degree. These infinities are not problematic for mathematical models but pose a serious problem for direct computer simulation.

Furthermore, it is not practical to implement PSMN as a classical computer simulation even for discrete complex systems due to the exponential increase in the complexity of the state vector. However quantum computers would be ideal for such simulations, because the exponential complexity arises when crossing from variable oriented methods to state space oriented methods and quantum processes are inherently state space oriented.

Aside from these practical limitations there are fundamental mathematical shortcomings with the PSMN approach.

2.8.1 $\sqrt{\text{NOT}}$ Logic Gate

The probabilistic nature of the PSMN state vector might seem to suit it to simulating quantum systems, however it is relatively easy to find quantum systems that the PSMN approach cannot simulate. The limitation with PSMN stems from the fact that it represents probabilities directly using real numbers, which means that the state vector cannot encode enough information to represent certain systems. A particular example will be used to highlight this shortcoming.

In a classical context a very simple logic gate is the NOT gate which maps $0 \rightarrow 1$ and $1 \rightarrow 0$, however in a quantum context we can define a variation on this, the

$\sqrt{\text{NOT}}$ gate. Given a pseudo-classical state, the $\sqrt{\text{NOT}}$ gate produces a mixed state, however another application of the $\sqrt{\text{NOT}}$ gate to this mixed state will result in the NOT of the original input. Thus $\sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}} = \text{NOT}$. Thus, in terms of probabilities:

$$\begin{pmatrix}1\\0\end{pmatrix} \xrightarrow{\sqrt{\text{NOT}}} \begin{pmatrix}0.5\\0.5\end{pmatrix} \xrightarrow{\sqrt{\text{NOT}}} \begin{pmatrix}0\\1\end{pmatrix} \tag{2.13}$$

$$\begin{pmatrix}0\\1\end{pmatrix} \xrightarrow{\sqrt{\text{NOT}}} \begin{pmatrix}0.5\\0.5\end{pmatrix} \xrightarrow{\sqrt{\text{NOT}}} \begin{pmatrix}1\\0\end{pmatrix} \tag{2.14}$$

But how is it that in (2.13) the mixed state transforms to one state whilst in (2.14) it transforms to another? This problem can be recast into a different form. Consider a transformation that maps a state s from $1 \to 0 \to -1$ as well as $-1 \to 0 \to 1$. If this operation is performed in one dimension, once $s = 0$ there is not enough information to specify whether the next state should be $s = 1$ or $s = -1$. However, if we consider a point in two dimensions the extra dimension encodes the required information.

If the s value is the projection of the point onto the x axis then the extra information encoded by the y axis allows us to implement the transformation as a simple rotation about the origin (fig. 2.3).

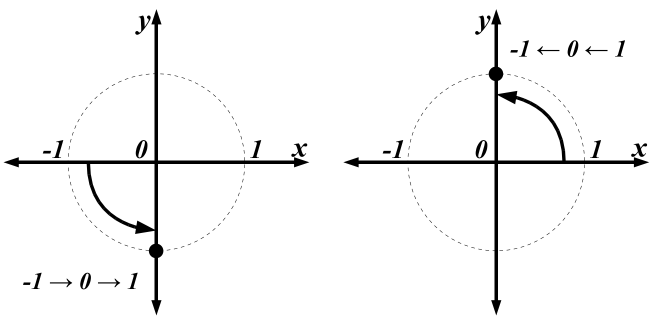

Figure 2.3: *Transformation in two dimensions.*

We could try explicitly using a hidden variable in the state vector but it turns out to be more effective to incorporate the extra information within the variables themselves. This involves shifting to the complex ring $\mathbb{C}$ and using variables of the form $z = x + iy$ where $i = \sqrt{-1}$. Then we can simulate systems such as the $\sqrt{\text{NOT}}$ gate.

By shifting to the complex ring we develop a quantum approach. The $\sqrt{\text{NOT}}$ gate example will be examined further in sec. 3.3, once the fundamentals of the complex ring have been introduced.

2.8.2 Group Algebra

There is a subtler problem with PSMN, which is also related to the need for complex values. To examine this let us introduce the concept of matrix spaces, which are similar to vector spaces only involving matrices rather than vectors. The set of all possible $N \times N$ matrices over a ring defines a matrix space. There exists a basis set of basis matrices such that all matrices within that space can be expressed as a linear combination of the basis matrices.

In the case of 2×2 matrices where $m_{ij} \in \mathbb{R}$ there are four basis matrices

$$\begin{array}{cc} b_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & b_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ b_2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} & b_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \end{array} \tag{2.15}$$

which form a basis set, from which any matrix M can be represented as a linear combination of the b_n matrices. From this we can calculate the matrix specified by a set of coefficients.

$$M = \sum_{n=0}^{3} c_n \cdot b_n \;=\; c_0 \cdot b_0 + c_1 \cdot b_1 + c_2 \cdot b_2 + c_3 \cdot b_3 \tag{2.16}$$

where the c_n are real coefficients. Furthermore,

$$M = \begin{pmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{pmatrix} = \begin{pmatrix} c_0 + c_3 & c_1 - c_2 \\ c_1 + c_2 & c_0 - c_3 \end{pmatrix} \tag{2.17}$$

from which we can calculate any particular matrix element from the coefficients. Also

$$\begin{array}{cc} c_0 = \frac{1}{2}(m_{00} + m_{11}) & c_1 = \frac{1}{2}(m_{01} + m_{10}) \\ c_2 = -\frac{1}{2}(m_{01} - m_{10}) & c_3 = \frac{1}{2}(m_{00} - m_{11}) \end{array} \tag{2.18}$$

from which we can calculate the coefficients that correspond to any given matrix.

Now consider the matrix group that forms under the operation of multiplication by the transpose. The transpose of a matrix changes each row to a column and each column to a row.

Thus $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \rightarrow \begin{pmatrix} a & c \\ b & d \end{pmatrix}$.

Below are listed all possible pairs of basis matrices under multiplication by the transpose.

$$\begin{array}{llll} b_0 b_0^T = b_0 & b_0 b_1^T = b_1 & b_0 b_2^T = -b_2 & b_0 b_3^T = b_3 \\ b_1 b_0^T = b_1 & b_1 b_1^T = b_0 & b_1 b_2^T = -b_3 & b_1 b_3^T = b_2 \\ b_2 b_0^T = b_2 & b_2 b_1^T = -b_3 & b_2 b_2^T = b_0 & b_2 b_3^T = b_1 \\ b_3 b_0^T = b_3 & b_3 b_1^T = -b_2 & b_3 b_2^T = b_1 & b_3 b_3^T = b_0 \end{array} \tag{2.19}$$

From the diagonal terms we see that $b_i^2 = b_0$ for all i thus the norm $\mathcal{N} = b_0^2 + b_1^2 + b_2^2 + b_3^2 = 4b_0$, which is positive definite [4]. This means that the matrix space has a well defined measure of distance or length ($\sqrt{\mathcal{N}}$) and any vector with zero length also has a zero norm. A classic example of a positive definite norm is Pythagoras' theorem where the length of a vector $l = \sqrt{x^2 + y^2}$.

However there is a problem with this group, which can be discerned by isolating the terms $b_0 b_2^T = -b_2$ and $b_2 b_0^T = b_2$. For a well formed algebra it is necessary that the identity matrix commutes with all other matrices. For example, in the group formed from real numbers under the operation of multiplication we find that $1 \cdot x = x \cdot 1$. However in this group of 2×2 matrices under the operation of multiplication by transpose we find that the commutation requirement is met by all matrices except for b_2. The two terms that we isolated are in fact anti-commutative because $b_0 b_2^T = -(b_2 b_0^T)$. This arises because $b_2^T = -b_2$. In order to achieve a positive definite algebra where the identity matrix is commutative we will have to change the matrix $b_2 \rightarrow ib_2$ where $i = \sqrt{-1}$ and also change the operation to multiplication by the adjoint (conjugate transpose). We will do this later in sec. 3.1.2 in the lead up to a quantum mechanical modelling methodology.

2.8.3 Complementarity

Another limitation of the PSMN approach is that it cannot represent the phenomenon of complementarity[1].

In PSMN all states that a system may exhibit are explicitly encoded within the

[1] Complementarity is the situation where a system can exhibit mutually exclusive observable forms depending on how it is observed. For instance, many physical systems can be observed as either a particle or a wave but never both at the same time. Each perspective arises from a different way of observing the system.

state vector thus any system state can simply be read off from the state vector. However numerous experiments, most notably the Stern-Gerlach experiment [5] prove that reality is more subtle than this. If two states are complementary then the more accurately we measure one of them the less we can know about the other. This leads to the Heisenberg uncertainty principle.

In the case of the Stern-Gerlach experiment a beam of silver atoms is passed through a magnetic field where the atoms are deflected depending on their magnetic dipole moment, thus the beam splits into two well defined beams. The orientation of the magnetic field can be changed to measure the dipole moment in different directions. If the beam is fired along the y axis then the field can measure dipole moments along either the x or z axes. If we first split the beam along the z axis and block the beam with $-z$ polarisation so that only $+z$ polarised atoms pass through, we can then measure this beam along the x axis producing two beams with $\pm x$ polarisation. If we then measure either of these beams along the z axis we will find that it consists of both $+z$ and $-z$ polarised atoms even though the $-z$ polarised atoms had been previously filtered out. Thus the polarisation of the magnetic dipole moment of an atom is not objectively encoded in the state of the atom. The act of measuring along the x axis put the z polarisation state into a mixed state of both $+z$ and $-z$ polarisation states. We will examine this experiment in more detail later in the context of quantum measurement (sec. 3.2.3)

E.C. Kemble described these quantum measurement phenomena by saying:

> "We have no satisfactory reason for ascribing objective existence to physical quantities as distinguished from the numbers obtained when we make the measurements which we correlate with them. There is no real reason for supposing that a particle has at every moment a definite, but unknown, position which may be revealed by a measurement of the right kind, or a definite momentum which can be revealed by a different measurement. On the contrary, we get into a maze of contradiction as soon as we inject into quantum mechanics such concepts as carried over from the language and philosophy of our ancestors...It would be more exact if we spoke of 'making measurements' of this, that, or the other type instead of saying that we measure this, that, or the other 'physical quantity'." [6]

An observation produces an observable, not the system state which is imperceptible, so to use observables to represent system states within a system model produces a model that is only of our perception of the appearances. To do so has been common practice throughout the history of simulation but this practice is subtly flawed. This issue is intimately related to naïve realism, which is an often unconsciously adopted epistemology (theory of knowledge) that asserts that systems fundamentally exist in the manner that they appear to exist [7]. For instance, a red object is believed to actually *be* red rather than just be a complex system which when observed produces a cognitive response that is experienced as redness. Similarly a particle is believed to actually *have* a location rather than just have some state which when observed produces an observed location. From the four aspects of virtual reality (discussed in sec. 1.2.7) it is clear that ultimately, it is naïve realism that causes virtual systems to perceive themselves as individual entities within a 'physical' universe.

Naïve realism is a fundamentally flawed epistemology, as we will later see in the chapter on quantum mechanics. Naïve realism is subtly problematic when attempting to simulate systems. This issue was touched upon earlier in regards to virtual systems within a linear algebraic context (sec. 1.2.7, pg 14). Of particular relevance here, the first and last aspects of a virtual system that were mentioned are:

1. It has a state that is represented by information within the state vector.

4. It has an observable form or *appearance*, which arises when its state information flows through an input interface and is perceived by a system's experiential process.

Thus the system state and the observable appearance are distinct. This applies not only to virtual systems within a simulation but also to physical systems, thus leading to the distinction between *wavefunction* and *observable* in quantum mechanics. Hence, to be accurate, the observable form of a system cannot be used to model the state of a system.

By addressing this issue, as well as the other limitations of PSMN, we enter the quantum realm where it becomes clear that objective reality is imperceptible and the observables that we perceive depend on how we look.

Chapter 3

Quantum Simulation (QSMN)

In this chapter we extend PSMN to overcome its limitations and thereby derive the mathematical foundations of quantum mechanics. We will encounter wavefunctions, Pauli matrices, unitary evolution operators, Hermitian observable operators, the Schrödinger equation of motion and complementary measurements on a quantum system. We also examine the $\sqrt{\text{NOT}}$ gate example and simulate a simple quantum computer. By this stage the rest is standard quantum mechanics which is discussed at length elsewhere.

Here we shall derive quantum mechanics in finite dimensional vector spaces. This is extended to the infinite dimensional Hilbert space elsewhere [8, 9].

3.1 Algebra on a Complex Ring

We are switching to the complex ring $\mathbb{C}$ thus we are considering variables of the form $z = x + iy$ where $x, y \in \mathbb{R}$ and $i = \sqrt{-1}$. It can be useful to think of these variables as a coordinate in a two dimensional vector space. Thus the basis vectors are 1 and i, whilst the coefficients are x and y. This two dimensional space is called the Argand plane.

3.1.1 State Vector → Wavefuntion

The state vector no longer represents the probabilities directly, instead the probability for the i^{th} state is $v_i \cdot v_i^*$ where v_i^* is the conjugate[1] of v_i. Thus

$$\begin{aligned} z \cdot z^* &= (x+iy)\cdot(x-iy) \\ &= x^2+y^2+ixy-ixy \\ &= x^2+y^2 \end{aligned} \tag{3.1}$$

This complex form of the state vector is identical to the quantum state vector, which represents the *wavefunction* or quantum state of a system. Rather than label it as a vector V we now use ψ or $|\psi\rangle$ to indicate that it is a wavefunction.

3.1.2 Complex Matrices

As mentioned earlier in regards to group algebra (pg 31), to overcome the limitations of PSMN and achieve a coherent matrix group algebra we need to adopt a complex ring $\mathbb{C}$ and use the operation of multiplication by the adjoint (conjugate transpose). This involves changing the b_2 basis matrix in (2.15). Let us explore what this entails.

For the case of 2×2 matrices where $m_{ij} \in \mathbb{C}$ there are four basis matrices

$$\begin{aligned} \sigma_0 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \sigma_2 &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \end{aligned} \tag{3.2}$$

that form a basis set, from which any complex matrix M can be represented as a linear combination of the σ_n matrices. In quantum mechanics σ_1, σ_2 and σ_3 are called the *Pauli matrices* and σ_0 is the identity matrix.

$$M = \sum_n z_n \cdot \sigma_n \;=\; z_0 \cdot \sigma_0 + z_1 \cdot \sigma_1 + z_2 \cdot \sigma_2 + z_3 \cdot \sigma_3 \tag{3.3}$$

where the z_n are complex coefficients. From this we can calculate the matrix specified by a set of coefficients. Furthermore,

$$M = \begin{pmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{pmatrix} = \begin{pmatrix} z_0 + z_3 & z_1 - iz_2 \\ z_1 + iz_2 & z_0 - z_3 \end{pmatrix} \tag{3.4}$$

[1] The conjugate of i is $-i$, thus if $z = x + iy$ then $z^* = x - iy$.

from which we can calculate any particular matrix element from the coefficients. Also

$$
\begin{array}{ll}
z_0 = \frac{1}{2}(m_{00} + m_{11}) & z_1 = \frac{1}{2}(m_{01} + m_{10}) \\
z_2 = i\frac{1}{2}(m_{01} - m_{10}) & z_3 = \frac{1}{2}(m_{00} - m_{11})
\end{array}
\tag{3.5}
$$

from which we can calculate the coefficients that correspond to any given matrix.

Now consider the matrix group that forms under the operation of multiplication by the adjoint. Below are listed all possible pairs of basis matrices under multiplication by the adjoint.

$$
\begin{array}{llll}
\sigma_0\sigma_0^\dagger = \sigma_0 & \sigma_0\sigma_1^\dagger = \sigma_1 & \sigma_0\sigma_2^\dagger = \sigma_2 & \sigma_0\sigma_3^\dagger = \sigma_3 \\
\sigma_1\sigma_0^\dagger = \sigma_1 & \sigma_1\sigma_1^\dagger = \sigma_0 & \sigma_1\sigma_2^\dagger = i\sigma_3 & \sigma_1\sigma_3^\dagger = -i\sigma_2 \\
\sigma_2\sigma_0^\dagger = \sigma_2 & \sigma_2\sigma_1^\dagger = -i\sigma_3 & \sigma_2\sigma_2^\dagger = \sigma_0 & \sigma_2\sigma_3^\dagger = i\sigma_1 \\
\sigma_3\sigma_0^\dagger = \sigma_3 & \sigma_3\sigma_1^\dagger = i\sigma_2 & \sigma_3\sigma_2^\dagger = -i\sigma_1 & \sigma_3\sigma_3^\dagger = \sigma_0
\end{array}
\tag{3.6}
$$

Where † represents the adjoint or conjugate transpose. From the diagonal terms we see that $\sigma_i^2 = \sigma_0$ thus the norm $N = \sigma_0^2 + \sigma_1^2 + \sigma_2^2 + \sigma_3^2 = 4\sigma_0$, which is positive definite. From the top row and left column we see that the identity matrix commutes with all matrices. In particular the terms that were previously problematic $\sigma_0\sigma_2^\dagger = \sigma_2$ and $\sigma_2\sigma_0^\dagger = \sigma_2$ now commute because $\sigma_2^\dagger = \sigma_2$.

Thus we not only have a matrix group with a well formed algebra but we have also derived the Pauli matrices which are used to represent the spin angular momentum or magnetic dipole moment of particles such as electrons, protons, atoms and so on. Spin angular momentum is analogous to the angular momentum of a top that is spinning in place. It is these matrices that explain the behaviour observed by the Stern-Gerlach experiment [10], which we will examine in detail in sec. 3.2.3.

3.2 Matrix Mechanics

3.2.1 Unitary Matrices

Matrices can represent many different operations, not just stepping a simulation forwards or back in time[2]. For instance they are also used to represent rotations. In both of these cases we need to have $M \cdot M^\dagger = I$, the identity matrix. For instance, a rotation followed by a reverse rotation should cancel out, just as a step forward in time followed by a step back should cancel out. Therefore the requirement that $M \cdot M^\dagger = I$ is a fundamental requirement of coherent matrix operations. This requirement implies that $M^\dagger = M^{-1}$ the inverse of M. A matrix that has this property is called a unitary matrix. A feature of unitary matrices is that both rows and columns are normalised. Hence they preserve the normalisation of the state vector when stepping a simulation both forwards and backwards. Likewise for both rotations and inverse rotations.

When a unitary matrix operates on a state vector in order to simulate a quantum system, such a matrix is called a *unitary evolution operator* $U(t) = U^t$. Thus $\psi(t) = U(t) \cdot \psi(0) = U^t \cdot \psi(0)$. This is logically equivalent to our earlier iterative equation $V_t = M^t \cdot V_0$.

When a unitary matrix is used to perform a rotation of bases, this is called a *unitary rotation matrix*. It is often seen in the form $M^\dagger O M$ where O is a Hermitian operator[3] that extracts some observable measurement from a state vector and the unitary matrix M translates the state vector into a desired basis set in order for the measurement to be made. This explanation has been expressed in the Schrödinger picture where it is the state vectors that evolve and the measurement operators are static. However it can also be interpreted in the Heisenberg picture where it is the measurement operators that evovle and the state vectors are static. We can switch between pictures depending on whether we use the unitary matrices for either evolution or rotation. For instance, an evolving measurement operator evolves according to $O(t) = U^\dagger(t) O(0) U(t)$.

3.2.2 Equation of Motion

It is possible to express a unitary operator as a function of a Hermitian operator. One notable form is $U = e^{iH}$ where H is Hermitian. This exponential form is particularly important in quantum mechanics because it leads us to an *equation*

[2] To obtain a time reversed simulation we use the adjoint of the causal matrix, thus every row becomes a column and vice versa. Hence every input interface becomes an output interface and vice versa. This means that information flows from input to output and the information process runs in reverse, thus the simulation runs in reverse.

[3] A Hermitian operator is self- adjoint, i.e. $O^\dagger = O$.

of motion for quantum systems, which describes the evolution of the state vector with respect to time. The outline of the following calculation was drawn from [8].

If we express the unitary evolution operator in the form

$$U(t) = e^{-itH/\hbar} \tag{3.7}$$

where H is the Hamiltonian (which describes the energetic dynamics of the system) and $\hbar = \frac{h}{2\pi} \approx 1.0546 \times 10^{-27}\text{g}\cdot\text{cm}^2/\text{s}$ where h is Plancks constant[4]. Then consider a wavefunction of the form

$$\psi(t) = U(t) \cdot \psi(0) \tag{3.8}$$

then consider the dependence of $\psi(t)$ on time, i.e. the rate of change of $\psi(t)$. By differentiation we get

$$\frac{\partial \psi(t)}{\partial t} = \frac{\partial U(t)}{\partial t}\psi(0) \tag{3.9}$$

Using (3.7) and the fact that $\frac{\partial}{\partial x}e^y = e^y \frac{\partial y}{\partial x}$ we get

$$\frac{\partial U(t)}{\partial t} = U(t)\frac{\partial}{\partial t}\left(\frac{-itH}{\hbar}\right) = \frac{-iH}{\hbar}U(t) \tag{3.10}$$

Substituting this into (3.9) gives

$$\frac{\partial \psi(t)}{\partial t} = \frac{-iH}{\hbar}U(t)\psi(0) \tag{3.11}$$

Using (3.8) and multiplying both sides by $i\hbar$ gives

$$i\hbar\frac{\partial \psi(t)}{\partial t} = H\psi(t) \tag{3.12}$$

This is the Schrödinger equation of motion, which is as fundamental to quantum mechanics as Newton's equations of motion are to classical physics.

3.2.3 Quantum Measurement

In the PSMN approach the measurement of the state of a system was trivial because the states were directly represented in the state vector. However in the quantum case things are a little more complex.

[4] Planck's constant describes the *quantisation* of the physical universe, which is analogous to the pixelation of a digital image.

If a measurement operator O is applied to a state vector ψ and results in a real valued multiple of the state vector $O \cdot \psi = k \cdot \psi$ where $k \in \mathbb{R}$ then the values k are eigenstates and the state vector is an eigenvector of the operator. This corresponds to a measurement of a state that happens to be directly represented by the state vector. A measurement of this kind leaves the state vector unchanged.

However there are other *bases* or sets of states that can be measured from the same state vector. Measurement of these states changes the state vector.

To illustrate quantum measurement of different bases we will examine the Stern-Gerlach experiment mentioned earlier (sec. 2.8.3), this example is drawn from [10]. The Pauli matrices $\sigma_1, \sigma_2, \sigma_3$ from equation (3.2) function as measurement operators on the spin state of a particle so from here on we will consider them to represent measurable quantities. The two states in the wavefunction represent spin up and spin down states and the three Pauli matrices measure the spin in the three different axes, x, y and z corresponding to $\sigma_1, \sigma_2, \sigma_3$. The spin angular momentum is measured by $\frac{1}{2}\hbar\sigma_i$ where $i = 1, 2, 3$ and the magnetic dipole moment is measured by $\mu\sigma_i$ where, for example, $\mu \approx -9.27 \times 10^{-21}\text{g}\cdot\text{cm}^2/\text{s}^2\cdot\text{gauss}$ for an electron and $\mu \approx 1.4 \times 10^{-23}\text{g}\cdot\text{cm}^2/\text{s}^2\cdot\text{gauss}$ for a proton. We will be considering the magnetic moments.

Notice that the spins in the three directions are not represented independently, but instead by a single two state wavefunction, however each of the three spin states in the three axes can be measured from that wavefunction. Because there is only enough information to fully specify one of these spin states, although each can be measured, they can only be measured one at a time.

For simplicity we leave out the μ value and just use the Pauli matrices. If there is a magnetic field that varies in space it induces a force that accelerates the particle in the direction of the changing magnetic field. The force is proportional to the projection of the magnetic moment in the direction of the field. This provides a way to measure the projections of the magnetic moment in different directions.

It is this force that causes the beam to split in the Stern-Gerlach experiment. The fact that the beam splits in two indicates that there are only two possible values of magnetic moment in each direction. In each direction $\sigma_i^2 = 1$. Because the square of the quantity has value 1 the quantity itself can have only the values $+1$ and -1 therefore the magnetic moments have only the values $+\mu$ and $-\mu$ in each direction. There are no intermediate values so the beam splits cleanly in two.

To determine the *mean value* of a quantity we multiply each state by its probability and sum them, i.e. $\langle\sigma_i\rangle = (-1) \cdot p(-1) + (1) \cdot p(1)$. If $p(-1) = 1$ and $p(1) = 0$ then $\langle\sigma_i\rangle = -1$. If $p(-1) = 0$ and $p(1) = 1$ then $\langle\sigma_i\rangle = 1$. If

$p(-1) = p(1) = 0.5$ then $\langle \sigma_i \rangle = 0$ (mixed state).

These mean values are not independent but are constrained by the inequality $\langle \sigma_1 \rangle^2 + \langle \sigma_2 \rangle^2 + \langle \sigma_3 \rangle^2 \leq 1$ (see [10] for the proof). Thus if $\langle \sigma_i \rangle = 1$ or -1 then the other two mean values must be 0, hence in a mixed state. Thus if one is defined the other two are undefined.

Thus, in the Stern-Gerlach experiment, the atoms were first filtered to only allow through spin up particles along the z axis, i.e. $\langle \sigma_3 \rangle = 1$. However as soon as the beam was split along the x axis this z-axis information was erased because once we measure $\langle \sigma_1 \rangle = 1$ or -1 then $\langle \sigma_3 \rangle = 0$ thus there is an equal likelihood of later measuring either spin up or spin down along the z axis.

From this example we see that the state of a quantum system is not directly encoded in the state vector, instead an abstract state is encoded, which must then be measured by a measurement operator. If the observables are complementary then it is impossible to accurately measure both at the same time, there simply isn't enough information to define both at the same time.

This complementarity not only applies to spin angular momentum along perpendicular axes but also to position/ momentum, energy/ time and many other observables. The phenomenon of complementarity leaves no doubt that the state of a system is imperceptible and the observables that we perceive and measure are produced in the moment of observation and do not inhere in the systems themselves.

It is these factors that have lead quantum physics to declare that:

> " "[W]e have to give up the idea of [naïve] realism to a far greater extent than most physicists believe today." (Anton Zeilinger)... By realism, he means the idea that objects have specific features and properties - that a ball is red, that a book contains the works of Shakespeare, or that an electron has a particular spin... for objects governed by the laws of quantum mechanics, like photons and electrons, it may make no sense to think of them as having well defined characteristics." [11]

It is not just particles and atoms that are governed by quantum mechanics either:

> "Quantum mechanics is increasingly applied to larger and larger objects. Even a one-ton bar proposed to detect gravity waves must be analysed quantum mechanically. In cosmology, a wavefunction for the whole universe is written to study the Big Bang. It gets harder today to nonchalantly accept the realm in which the quantum rules apply as somehow not being physically real... " Quantum

mechanics forces us to abandon naïve realism". And leave it at that." [12]

3.3 Simulating Quantum Systems

To illustrate how to simulate a quantum system we will consider the example of a quantum computer that computes the square of the $\sqrt{\text{NOT}}$ function on a single *qubit* (two state wavefunction). This will also serve as an illustration of quantum computing. The example is drawn from [13].

Recall that given a pseudo-classical state the $\sqrt{\text{NOT}}$ gate produces a mixed state, however another application of the $\sqrt{\text{NOT}}$ gate to this mixed state will result in the NOT of the original input. Thus $\sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}} = \text{NOT}$. The causal matrix that describes the $\sqrt{\text{NOT}}$ gate is

$$\sqrt{\text{NOT}} = \begin{pmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{pmatrix} \tag{3.13}$$

This causal matrix is unitary

$$\sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}}^{\dagger} = \begin{pmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{pmatrix} \cdot \begin{pmatrix} \frac{1-i}{2} & \frac{1+i}{2} \\ \frac{1+i}{2} & \frac{1-i}{2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{3.14}$$

And it has the desired behaviour

$$\begin{aligned} \sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}} &= \begin{pmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{pmatrix} \cdot \begin{pmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \text{NOT} \end{aligned} \tag{3.15}$$

Although this causal matrix describes *what* will happen overall it doesn't describe *how* the quantum system will evolve as the process unfolds, to do this we need the unitary evolution operator. To determine the unitary evolution operator we can calculate it using Schrödinger's equation of motion (3.12) and the Hamiltonian for a general quantum computer, which was discovered by Richard Feynman.

First we start with a circuit level description of the computation that we wish to implement, in this case it involves two quantum logic gates being applied to one qubit. In general we can divide the overall action of the unitary operator into the application of k separate operators for each of k logic gates that operate on m qubits, thus it can be written as $A_k \cdot A_{k-1} \cdot \ldots \cdot A_0$.

We want a Hamiltonian H such that $U(t) = e^{-itH/\hbar} = \left(e^{-iH/\hbar}\right)^t$ results in a unitary evolution operator that implements the dynamical evolution described by $A_k \cdot A_{k-1} \cdot \ldots \cdot A_0$.

Feynman found that by augmenting the circuit with $k+1$ extra qubits, which serve as a program counter[5], the Hamiltonian could be expressed as

$$H = \sum_{i=0}^{k-1} c_{i+1} \cdot a_i \cdot A_{i+1} + (c_{i+1} \cdot a_i \cdot A_{i+1})^\dagger \tag{3.16}$$

Where c and a are creation and annihilation operators. The matrix

$$c = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \tag{3.17}$$

is called a creation operator because it converts a 0-state to a 1-state and a 1-state to the null state. The matrix

$$a = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{3.18}$$

is called an annihilation operator because it converts a 1-state to a 0-state and a 0-state to the null state. To produce a creation or annihilation operator that operates on the i$^{\text{th}}$ qubit (i.e. c_i or a_i) we can use a matrix direct product [14] with identity matrices in all positions except the i$^{\text{th}}$, where there is either a c or a. For instance $c_2 = I \otimes c \otimes I \otimes I$.

The effect of this Hamiltonian is such that the creation and annihilation operators move the counter backwards and forwards and apply the corresponding gate operator only when the counter is in the correct position. Thus we can measure the counter qubits and when the final qubit is set we know that the computation has completed and the answer (or superposition of answers) can be found in the output qubits.

In our case there are $k = 2$ logic gates so we need $k+1 = 3$ counter qubits, thus we need four cubits all up, including the single in / out qubit that the logic gates operate on. Thus the memory register of the quantum computer can be visualised as $\begin{pmatrix} C_0 & C_1 & C_2 & M \end{pmatrix}$ where C_i represents the counter qubit for the i$^{\text{th}}$ step and M represents the memory element that holds both the input state and the resulting output state.

For our particular example there are two identical gate operators $A_1 = A_2$, both of which are $\sqrt{\text{NOT}}$ gates as shown in (3.13). However the causal matrix shown operates only on a single qubit, hence we need to use a direct product in a similar fashion to the creation and annihilation operators. Given the structure

[5] The use of a program counter can also be applied to general SMN simulations.

of the quantum memory register we want the gate operators to operate on the last of the qubits, hence $A_1 = A_2 = I \otimes I \otimes I \otimes \sqrt{\mathrm{NOT}}$. Now that we have the creation and annihilation operators as well as the gate operators we can calculate the Hamiltonian using (3.16), which produces the matrix

$$H = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{a} & \mathbf{b} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{b} & \mathbf{a} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{b} & \mathbf{a} & 0 & 0 & 0 & 0 & \mathbf{a} & \mathbf{b} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{a} & \mathbf{b} & 0 & 0 & 0 & 0 & \mathbf{b} & \mathbf{a} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{a} & \mathbf{b} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{b} & \mathbf{a} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{b} & \mathbf{a} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{a} & \mathbf{b} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{b} & \mathbf{a} & 0 & 0 & 0 & 0 & \mathbf{a} & \mathbf{b} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{a} & \mathbf{b} & 0 & 0 & 0 & 0 & \mathbf{b} & \mathbf{a} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{b} & \mathbf{a} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{a} & \mathbf{b} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{3.19}$$

where $a = \frac{1+i}{2}$ and $b = \frac{1-i}{2}$. Given this Hamiltonian we can calculate the unitary evolution operator using $U(t) = \left(e^{-iH/\hbar}\right)^t$ and the power series expansion

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \tag{3.20}$$

where ! represents the factorial[6]. We then consider each matrix element and use the power series expansions of *cos* and *sin*. This produces the unitary evolution matrix.

[6] $q\,! = q \cdot (q-1) \cdot \ldots \cdot 1$

$$U(t) = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \beta & 0 & \delta & \varepsilon & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \beta & \varepsilon & \delta & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \varepsilon & \delta & \alpha & 0 & 0 & 0 & \delta & \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \delta & \varepsilon & 0 & \alpha & 0 & 0 & \varepsilon & \delta & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & \delta & \varepsilon & 0 & \gamma & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & \varepsilon & \delta & \gamma & 0 & 0 & 0 \\
0 & 0 & 0 & \gamma & \varepsilon & \delta & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \gamma & 0 & \delta & \varepsilon & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \varepsilon & \delta & 0 & 0 & \alpha & 0 & \delta & \varepsilon & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \delta & \varepsilon & 0 & 0 & 0 & \alpha & \varepsilon & \delta & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & \varepsilon & \delta & \beta & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & \delta & \varepsilon & 0 & \beta & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}^t \tag{3.21}$$

where

$$\begin{aligned}
\alpha &= \cos\sqrt{2} \\
\beta &= \tfrac{1}{2}(1+\alpha) \\
\gamma &= \tfrac{1}{2}(-1+\alpha) \\
\delta &= \tfrac{1}{2\sqrt{2}}(1-i)\sin\sqrt{2} \\
\varepsilon &= \tfrac{1}{2\sqrt{2}}(-1-i)\sin\sqrt{2}
\end{aligned} \tag{3.22}$$

Now we produce an initial state vector from the direct product of four qubits, $|\psi(0)\rangle = |1\rangle \otimes |0\rangle \otimes |0\rangle \otimes |q\rangle = |100q\rangle$ where the $|1\rangle$ qubit sets the first counter to the 1-state, the $|0\rangle$ qubits set the rest of the counters to the 0-state and the final $|q\rangle$ qubit sets the input qubit to the desired input-state, which could be a superposition but here it is sensible to use either $|0\rangle$ or $|1\rangle$.

Then the state of the quantum memory register (state vector) at any future moment can be determined by $|\psi(t)\rangle = U(t) \cdot |\psi(0)\rangle$. Note that the time dependence of $U(t)$ raises the matrix to the power of t, thus $U(t) = U^t$. This is then applied to the initial state. We can also leave $t = 1$ and repeatedly apply the causal matrix to iterate the simulation forward in time in order to observe its dynamical behaviour. Let us set the input state to $|0\rangle$ thus the initial state is $|1000\rangle$ and we will compute $\sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}} \cdot |0\rangle = \text{NOT}|0\rangle$ which should produce the output state $|1\rangle$.

After the simulation has been running for a number of iterations the state vector will be in a mixed state representing a superpostion of all possible states. There are four binary qubits so there are $2^4 = 16$ possible states and corresponding to each state there is a probability. Eventually we will find this mixed state in a form similar to:

$\|0000\rangle : 0$	$\|0100\rangle : 0.04$	$\|1000\rangle : 0.02$	$\|1100\rangle : 0$
$\|0001\rangle : 0$	$\|0101\rangle : 0.04$	$\|1001\rangle : 0$	$\|1101\rangle : 0$
$\|0010\rangle : 0$	$\|0110\rangle : 0$	$\|1010\rangle : 0$	$\|1110\rangle : 0$
$\|0011\rangle : 0.9$	$\|0111\rangle : 0$	$\|1011\rangle : 0$	$\|1111\rangle : 0$

Where each state and its corresponding probability are listed.

Given the structure of the computational process, encoded in the Hamiltonian, most of the states are forbidden; hence there are many zero probabilities. For instance, only one counter qubit can be in a 1-state at any given moment.

Thus the probabilities shown above suggest that there is a probability of 0.02 that the memory register is still in the initial state. There is a probability of 0.04 that the first logic gate has been applied, leaving the output in a 0-state. There is a probability of 0.04 that the first logic gate has been applied, leaving the output in a 1-state. These two have equal probabilities implying that there is a probability of 0.08 that the second step has been applied thus leaving the output in a mixed state, as expected from one application of the $\sqrt{\text{NOT}}$ gate. Finally, there is a probability of 0.9 that the second logic gate has been applied resulting in a 1-state in the output.

If there was still significant probability for the initial $|1000\rangle$ state or the intermediate $|0100\rangle$ and $|0101\rangle$ states then the system would be allowed to evolve further. However once there is significant probability that the system is in its final state $|0011\rangle$ we can then read off the result of the computation.

This example has illustrated the relationship between the description of *what* happens (3.13) and the description of *how* it happens (3.21). These two are related via the Schrödinger equation of motion (3.12) and the Hamiltonian(3.16, 3.19). This has also served as an example of a simple quantum computation.

3.4 Continuous Quantum Simulation

We have thus far used system mechanics to derive the foundations of quantum mechanics for finite dimensional complex vector spaces, however in general quantum mechanics operates within an infinite dimensional complex vector space, called *Hilbert space*. The basic principles are the same however the introduction of infinities produces some subtleties. We will not examine this here since it is addressed elsewhere, for example [8, 9].

Chapter 4

Quantum and Classical Systems

Now that we have developed quantum system simulation we can further develop the classical approach and then examine how the quantum and classical system models relate to each other and what they imply about virtual systems.

4.1 Classical Simulation (CSMN)

Classical SMN is *non-linear* and isn't strictly *algebraic* however it still works according to the same general priniciples as the other varieties of SMN. Instead of using matrix / vector multiplication of the standard mathematical variety it uses *generalised* matrix / vector multiplication. To illustrate how this works lets consider the standard form as a special case of the generalised form. In the standard form we have

$$\begin{pmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} = \begin{pmatrix} (m_{00} \cdot v_0) + (m_{01} \cdot v_1) \\ (m_{10} \cdot v_0) + (m_{11} \cdot v_1) \end{pmatrix} \tag{4.1}$$

Note that for classical systems the state vector is not a probability distribution, instead it contains classical state variables in the same manner as the water tanks example (sec. 1.2).

Here we can identify two operations, firstly a pairwise operation $P \equiv () \cdot ()$ between a matrix element and a vector element, and secondly a row operation $R \equiv () + ()$ between all pairs resulting from a row of the causal matrix being applied to the state vector. Thus in the case of standard matrix / vector multiplication we see that P corresponds to scalar multiplication and R corresponds to summation. In general we have

$$\begin{aligned}
&\left[\begin{array}{c|cc} R_0 & P_{00} & P_{01} \\ R_1 & P_{10} & P_{11} \end{array}\right] \begin{pmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} \\
&= \begin{pmatrix} R_0\Big(P_{00}(m_{00}, v_0), P_{01}(m_{01}, v_1)\Big) \\ R_1\Big(P_{10}(m_{10}, v_0), P_{11}(m_{11}, v_1)\Big) \end{pmatrix}
\end{aligned} \tag{4.2}$$

So the standard matrix / vector multiplication expressed in this notation gives us

$$\begin{aligned}
&\left[\begin{array}{c|cc} \sum & \cdot & \cdot \\ \sum & \cdot & \cdot \end{array}\right] \begin{pmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} \\
&= \begin{pmatrix} (m_{00} \cdot v_0) + (m_{01} \cdot v_1) \\ (m_{10} \cdot v_0) + (m_{11} \cdot v_1) \end{pmatrix}
\end{aligned} \tag{4.3}$$

In the general case the causal matrix and state vector can be treated as a data structure whilst the R and P operations can be treated as general functions that operate on that data.

P can be any function that takes a matrix element and a vector element as input then returns a single output of any kind. R can be any function that takes the outputs of the P functions as its input then returns a single vector element. The type of the inputs to P are constrained and the type of the output from R is constrained but what happens in between is completely unconstrained. Any type of operation can be performed, ranging from a simple mathematical function to a complex algorithm or computer program.

Furthermore, the vector elements can be any arbitrary data from numerical quantities to binary data such as strings, bitmaps or anything. The matrix elements can be any arbitrary data or function so long as the P functions know how to combine it with the corresponding vector element.

A useful variation is to let the matrix elements be a pair of functions I_{ij} (input filter) and O_{ij} (output filter), then the pairwise function applies these two functions in succession, thus $P_{ij}(m_{ij}, v_j) \equiv I_{ij}(O_{ij}(v_j))$.

Here the vector element v_j represents the internal state of a system. The output filter $O_{ij}(v_j)$ represents that internal state being expressed through a particular interaction channel (thus systems can express themselves differently to different systems). The input filter $I_{ij}(O_{ij}(v_j))$ is an expression being interpreted by the receiving system.

Thus the internal state of each system is expressed then interpreted. Then a row function merges these interpreted expressions from many interaction channels into an overall interpretation of the system's perception of its environment. The output of the row function is the system's response to that perception, which is a new internal state for that system. This internal state is then expressed along different channels, whereon it is interpreted and responded to. Thus the information flows throughout the system network and animates the system dynamics.

This approach produces a model of general *massively parallel computational processes*. CSMN is an algorithm that can transform a single computational process into any number of virtual parallel computational processes. Furthermore, given that matrix algorithms are easily parallelisable any number of processors operating on shared memory can share the computational load, hence it can transform any number of processors into any number of virtual parallel processes. All of the parallelisation issues such as dead-locking and re-entrance are handled seamlessly by the CSMN algorithm.

Unlike the state space system models CSMN does not have an exponential increase in complexity, hence it is ideal for implementation on classical computers. The matrix approach has certain overheads however these can be largely avoided using sparse matrices so that only the non-zero matrix elements are represented.

Furthermore, rather than have an entire causal matrix operate on the vector in each iteration there is a more efficient approach that is called *energy flow* processing. For instance, if an input interface draws information from certain vector elements and these have not changed in the last iteration then there is no need to process that input interface again because the result will be the same as before. Thus when a vector element changes, the algorithm scans down the corresponding column of the causal matrix to see which systems depend on that state and only those systems are processed. This means that the computational load on the simulator is dependent on the amount of change occuring in each simulation moment, hence it depends on the flow of energy through the simulation. This results in a logical equivalence between virtual energy and computational load.

CSMN not only provides a method by which to simulate and analyse general computational processes, but also a way to implement them. This is a form of *system-oriented* software engineering. The system designer need only understand the target system and use any system modelling methodology such as UML, or any formal language such as C++, Java, OWL and so on. CSMN can also be given a GUI interface so that the system designer simply creates graphical system networks and assigns behaviours to the various components. The system model can then be converted into a CSMN model and directly simulated.

Thus the system designer doesn't need to understand a particular modelling language or programming language, any formal language that can specify the system will suffice. Once the system model is converted to CSMN form it can be analysed and tested using methods adapted from linear algebra. For maximum efficiency it can also then be converted into C code to avoid what little overhead there is from the CSMN algorithm itself.

An SMN system simulation is a closed system, however this can be augmented with *system wrappers*. These are virtual systems within the simulation that represent external systems such as software programs or procedures, onscreen display objects and any electronically controllable equipment. When data is written to the system wrapper it is relayed to the external system and when data is read from the system wrapper it is drawn from the external system. In this way external systems can participate in the virtual system dynamics and the virtual systems can participate in external dynamics.

For more information and the C source code of a prototype implementation (still in alpha development phase) see [15]. There are also previous versions that implement different system models such as a particle simulator / drawing program where the 'brushes' are systems of particles, cellular automata, a fluid drawing canvas and a drink machine [16]. There are also examples of mathematical models of relativistic particles, springs and logic gates [17].

4.2 Logical Equivalence

To illustrate the logical equivalence between quantum and classical system models lets consider the earlier example of the NAND / XOR logic circuit (from sec. 2.6).

In the quantum case we had

$$M \cdot V = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} p(00) \\ p(01) \\ p(10) \\ p(11) \end{pmatrix} \tag{4.4}$$

Its classical equivalent is

$$\left[\begin{array}{c|cc} \text{NAND} & \cdot & \cdot \\ \text{XOR} & \cdot & \cdot \end{array}\right] \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} = \begin{pmatrix} v_0 \text{ NAND } v_1 \\ v_0 \text{ XOR } v_1 \end{pmatrix} \tag{4.5}$$

where the pairwise function is $P_{ij} \equiv () \cdot ()$ and $m_{ij} = 1$, which selects the vector elements and presents them to the row function, which then applies a logical operation to them.

We could easily change the quantum causal matrix slightly to produce a simulation that did not correspond to any known logical operation. However we can represent the transformations between pseudo-classical states as a set of truth tables and then extend the set of logical operations to include all possible truth tables. The mixed states (superpositions) can be represented as probabilistic (stochastic) processes. In this way we can accommodate both the non-algorithmic variations and the mixed states. For instance, instead of using NAND, XOR and other recognised logical operations, we can enumerate all the possible truth tables and index them as Op_i then each row function can implement one of these Op_i operations and every quantum model will have a corresponding classical model.

Thus we see that for every quantum model there is a classical equivalent even though most of these employ non-standard logical operators and stochastic processes. Thus the two are logically equivalent algorithms for system simulation. The quantum approach is more fundamental, however the classical equivalent is closer to our common sense understanding of systems.

4.3 Virtual Systems

In the earlier discussion on the four aspects of virtual systems and virtual reality (sec. 1.2.7) we examined a simple classical model. Each system had interfaces that corresponded to a particular row and column. From the generalised model we see that systems control the output filters in their output interface (column) and the input filters in their input interface (row).

$$\begin{bmatrix} \ddots & ,\mathrm{O} & \cdots \\ \mathrm{I}, & \mathrm{I},\mathrm{O} & \mathrm{I}, \\ \vdots & ,\mathrm{O} & \ddots \end{bmatrix} \tag{4.6}$$

Thus a system can control how it expresses itself to others and how it interprets the expressions of others. It cannot control the expressions or interpretations of other systems. The only way it can influence others is by how it expresses itself (output filters) and by its own responses (changes in internal state), which depend on its interpretations (input filters).

This not only applies to *atomic systems* (systems with no sub-systems) but also to complex systems (systems with sub-systems). In the complex case there are several vector elements, rows and columns that relate to the system. These are conceptualised as a single complex vector element, row and column. Thus the same conclusions apply to both atomic and complex systems.

This analysis is of the classical case, however the logically equivalent quantum case leads to a different picture of a system. In the quantum model there is no longer a particular row and column that correspond to a system. Instead individual systems are represented solely by their component of the joint state and all operations are in terms of transitions between joint states.

In the quantum model there are no explicit individual entities. There is only the unified quantum system and its state transitions. It is only once we shift to the classical equivalent that the individual systems are explicitly represented. Due to the unified quantum foundation the interactions between the individual systems exhibit global coherence.

This suggests that fundamentally there is only a single unified system that undergoes state transitions. The quantum state vector represents a *multiverse* wherein all possible universal configurations are represented as potentialities. Thus there are no explicit systems or system interactions, there is only a unified whole that is represented by changing probability distributions within a multiverse. However the classical picture is logically equivalent, thus another way of looking at the situation is in terms of individual systems and system interactions.

Thus there are two ways of interpreting the same situation. Either everything is one and whole or all things are separate and individual. This apparent duality can be inferred from the same reality generative information process by interpreting it in different ways.

Chapter 5

Summary and Extension

To summarise what has transpired thus far, we started with simple observations about systems, developed a mathematical model of systems then extended this until it was coherent, thus deriving the foundations of quantum mechanics. We have also examined some of the principles of virtual reality. In doing so we have developed system mechanics and shown that quantum mechancs is a sub-domain of system mechanics that studies quantum systems.

This results in a system theoretic conceptual framework that provides a coherent foundation upon which key concepts can be developed further. Before moving on, there is more that can be said to extend the foundation that has been developed so far.

5.1 Non-Ergodic SMN

An ergodic system is one where the transition probabilities remain constant. For instance, if when a system is in state s_0 there is a probability $p(0 \to 1)$ that it changes to state s_1 then that causal relation does not change from one moment to the next. If the state vector represents the state of an entire universe and the laws of nature remain constant then this would be an ergodic system model. Such a system can be modelled by a single causal matrix that drives a cyclic evolutionary process of the state vector.

However, in general the systems that we wish to model are not an entire universe thus we may require different matrices to model their non-ergodic evolution. For instance, just because a piece of machinery functions smoothly at one point doesn't mean that it will always do so. The machinery may become worn, hence the causal framework that defines the causal relations between the sub-systems changes.

Furthermore, when simulating an entire universe there is a possibility that the laws of nature do in fact change, albeit rather slowly on the time-scale of the virtual systems and even if the laws of nature are constant there is the possibility that once one full evolutionary cycle has been completed using one causal matrix, another cycle will begin using a different causal matrix. In each of these cases the causal matrix must change in some causally determined manner so it is useful to enquire how a modelling methodology can accommodate a changing causal framework.

So far we have only considered the case where a single causal matrix operates on a state vector such that $V_{t+1} = M \cdot V_t$. The elements of the state vector change whilst the causal matrix and therefore the causal relations remain constant. Hence we have only considered ergodic systems thus far. However the SMN approach can be extended to simulate non-ergodic systems as well.

The way we do this is to use the same approach for simulating target systems to simulate SMN itself, thus SMN becomes the target system that is to be simulated by a higher-level SMN simulator. Hence an SMN process is nested within and SMN process. We will not explore these nested models in detail but a brief overview is provided.

First consider that the state vector V represents states of being and the causal matrix M represents ways of changing. Hence their combination $M \cdot V$ represents an event within the simulation; i.e. a transition from a particular state to another.

Earlier in (2.12) we saw a complete list of all possible $M \cdot V$ combinations within a deterministic simulation of two binary systems. This is a complete list of all possible events that can occur within the corresponding simulation.

Let the state vector E represent a probability distribution over this complete list of events, so each event has an associated probability, which is represented by a particular element of E. Then let the causal matrix C be a causal framework that determines how the occurance of one event results in the likelihood of other events in the next moment (iteration). Thus the causal matrix M may change from moment to moment thus the causal framework within the simulation may change.

An iteration of $C \cdot E$ produces a new E state vector. For simplicity lets assume that it is a pseudo-classical state vector although it need not be. A pseudo-classical E vector specifies a particular $M \cdot V$ pair which when iterated produces a new state vector V that defines the state of the next moment of existence for virtual systems. The next iteration of $C \cdot E$ produces a new $M \cdot V$ pair but the vector in this pair must be the same vector that resulted from the previous $M \cdot V$ iteration, only the M causal matrix can be different. This requirement places constraints on the form of the C causal matrix.

The C causal matrix represents how the causal framework changes over the life-cycle of a system, i.e. it is a causal framework over a causal framework. The E vector represents the initial state of a system, i.e. it is a state vector defining the initial events that trigger a cycle of events, which unfold into a system's life-cycle. Thus each $C \cdot E$ pair is a seed, which when iterated produces an entire evolutionary cycle of a non-ergodic system.

In the un-nested $M \cdot V$ case the states may be non-classical but the events themselves are classical in the sense that only one event ($M \cdot V$ iteration) can occur at a time. However if the E vector is non-classical we have a probability distribution over events, hence the events themselves are non-classical. In this case there will be a probabilitiy distribution over the resulting V vectors, which then influences the form of the new probability distribution over the E vector in line with the constraint on the form of C mentioned previously.

This *nested SMN* approach can be taken further although the results of this are highly speculative. In the case of simulating an entire universe the next level simulates how one universal cycle follows another.

The next level has a state vector X representing every possible $C \cdot E$ combination. There is also a causal matrix Y, such that $Y \cdot X$ determines how one universal cycle follows another. Once a $C \cdot E$ pair is selected this seed is iterated for a complete cycle before the next pair is selected. Thus there is a primary and secondary SMN process, the primary process selects a universal seed and initiates a secondary process that iterates it to completion, whereon the secondary process ceases and the primary process selects another seed and initiates another secondary process. If the X vector is pseudo-classical it selects only one $C \cdot E$ pair to be iterated thus there is only one universe however if it is non-classical then there may be a mixed state of multiple universes.

This could be taken further still but it is unclear what possible meaning could be associated with the deeper nested levels. The three level nesting of $M \cdot V$, $C \cdot E$ and $Y \cdot X$ can represent all states of being, ways of changing, events, evolving causal frameworks, universal life-cycles and universal successions.

5.1.1 Structural Nesting

There is another form of *nested* SMN, which is not related to the form just described. To distinguish between them we may call the above form *processual nesting* whilst the other form may be called *structural nesting.* In structurally nested SMN the elements of the causal matrix can themselves be matrices and the corresponding elements of the state vector can themselves be state vectors. This nesting can be many levels deep however it must at some point *bottom out* with scalar values as matrix and vector elements.

When a nested causal matrix is multiplied with a nested state vector the pair-wise terms $m_{ij} \cdot v_j$ are not a multiplication of scalars, they are another matrix / vector multiplication. This can be understood by visualising a system network where each node is itself a system network and each edge is a connection between multiple nodes within the sub-networks. First consider the system network Z, which contains two nodes z_0 and z_1 (fig. 5.1).

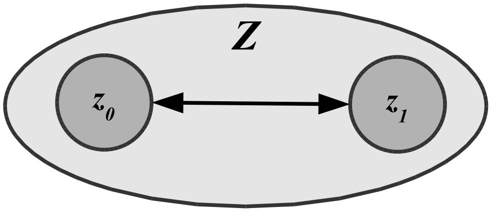

Figure 5.1: A simple system network.

However each of these nodes is itself a system network, which are labelled X and Y (fig. 5.2).

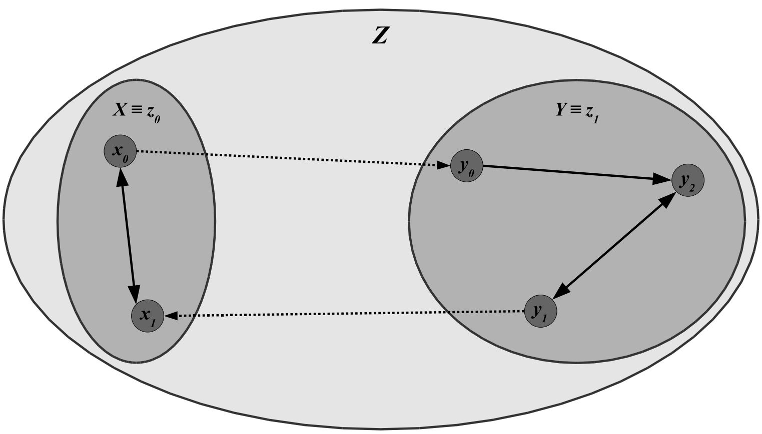

Figure 5.2: A structurally nested system network.

This nesting can be many levels deep. The fully expanded system network (containing all nested levels) can be represented as a single flat (non-nested) causal matrix and vector, which is logically equivalent.

In matrix notation the system network Z consists of two sub-networks X and Y, where $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $Y = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$.

These can be arranged within a larger causal matrix, which is logically equivalent to a flat causal matrix (the lines indicate the regions of the flat matrix that have been nested).

$$M_z = \begin{pmatrix} M_X & M_{X \leftarrow Y} \\ M_{Y \leftarrow X} & M_Y \end{pmatrix} \equiv \left(\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{array}\right)$$

The causal matrices M_X and M_Y represent the sub-networks X and Y. $M_{X \leftarrow Y}$ represents the connections to X from Y. $M_{Y \leftarrow X}$ represents the connections to Y from X.

Whether a causal matrix is flat or nested is irrelevant to the functioning of the simulation because both are logically equivalent. However it can be useful to use a modular causal matrix model where sub-system causal matrices and their interaction matrices are developed separately and then inserted into a super-system model.

Furthermore, the flat system model can be partitioned in different ways, thus producing different groups of different sub-networks. Hence the sub-system boundaries can be arbitrarily defined.

In general, a partition that maximises the connectivity within sub-systems and minimises the connectivity between sub-systems will be the most useful and the one that agrees most with our own perception of systems. This relates to the subject of meta system transition, joint probabilities, complex systems and the formation of 'objects'. These were discussed in sec. 2.5.1 where we found that objects are actually objects of perception and have no fundamental identity in themselves, they are an emergent property of coordinated sub-system interactions which produces the appearance of an individual entity.

5.2 Quantum Mechanics

5.2.1 Matrices or Waves

There are two logically equivalent mathematical formulations of quantum mechanics; matrix mechanics (developed by Heisenberg) and wave mechanics (developed by Schrödinger). In the early days of quantum mechanics both of these emerged at much the same time. There was much debate over which approach was preferable and people gravitated towards wave mechanics because the matrix mathematics was unfamiliar and the mathematics of waves was already familiar due to its application in classical physics. It was also believed that waves were more 'physical' than abstract information processes represented by matrices.

However it was later noticed by Max Born [18] that quantum waves were distinctly non-physical in the sense that they were not a physical wave travelling through something but instead they were waves of the complex square root of probabilites, which had to be squared in order to result in anything physically meaningful. They were therefore numbers that cycled in the manner of waves in order to participate in a purely computational process. Thus the 'physicallity' of wave mechanics was only superficial.

At the time physicists were very familiar with classical physics and had little experience with computers or computational processes, let alone of virtual reality. However, in light of these it becomes apparent that the matrix mechanics and quantum phenomena in general can be understood as computational processes that do not operate within space and time, instead space and time arise from the computational processes. This is the essence of the systems paradigm.

> "Richard Feynman developed an elegant model for describing the amplitude of the many waves involved in a quantum event, calculating the interference of all of these amplitudes, and using the final result to calculate a probability. However, Feynman disclaimed any insight into whatever physical process his system might be describing. Although his system achieved a result that was exactly and perfectly in accord with observed natural processes, to him it was nothing more than calculation. The reason was that, as far as Feynman or anybody else could tell, the underlying process itself was nothing more than calculation... A process that produces a result based on nothing more than calculation is an excellent way to describe the operations of a computer program. The two-step procedure of the Schrödinger equation and the Feynman system may be impossible to duplicate with physical systems, but for the computer it is trivial." [18]

The uncanny utility of mathematics in the sciences has been a mystery throughout the history of science, however if it turned out that the universe was in a sense the result of a calculation (i.e. a simulation) then it is obvious that mathematics would be the ideal way to study physical systems.

The superficially 'physical' waves, or any physicalist interpretation of quantum mechanics, cannot lead towards any great insight into the nature of quantum processes. Hence there have been many statements such as

> "The theory has, indeed, two powerful bodies of fact in its favour, and only one thing against it. First, in its favour are all the marvellous agreements that the theory has had with every experimental result to date. Second, and to me almost as important, it is a theory of astonishing and profound mathematical beauty. The one thing that can be said against it is that it makes absolutely no sense!" [19]

However the matrix mechanics approach can lead to deeper insight into the nature of quantum processes. Quantum matrix mechanics can be conceptually integrated with system mechanics and through this avenue the computational nature of quantum processes becomes apparent and so too the virtual nature of the physical universe. This explains why any physicalist interpretation is bound to be non-sensical and only a computational / systems interpretation can yield deeper insights.

Furthermore, via system theory it may be possible to extend quantum theory into the realm of macroscopic complex systems. With a naïve realist, materialist ontology it was impossible to do so, however it can be achieved by reconceptualising macroscopic objects as complex systems and by recognising the informatic nature of general systems and the quantum nature of the information processes that animate them. This could eventually result in a genuine theory of everything (TOE).

5.2.2 Cybernetic Interpretation

It is not just the mathematical formulation of quantum mechanics that suggests that reality has a computational nature, many of the properties of quantum systems are incomprehensible within a physicalist paradigm but are obvious and essential within a computational / systems paradigm.

Wave particle duality

In Young's double slit experiment it was found that light behaved both like a wave and a particle. After passing through two slits it formed interference patterns on a screen exactly like waves. However when the intensity of light was reduced to the point that there could only be one photon at a time passing through the slits, that single photon still interferred with itself as if it was a wave.

> "Investigating the mechanics of this process turns out to be impossible, for the reason that whenever we try to observe or otherwise detect a wave we obtain, instead, a particle. The very act of observation appears to change the nature of the quantum unit… the 'wave function' is somehow 'collapsed' during observation, yielding a 'particle' with measurable properties. The mechanism of this transformation is completely unknown and, because the scientifically indispensable act of observation itself changes the result, it appears to be intrinsically and literally unknowable… As John Gribbin puts it, "nature seems to 'make the calculation' and then present us with an observed event."" [18]

Ontological Sameness

All macroscopic objects possess some individual identity in the sense that in principle they are able to be distinguished. This property is called *primitive thisness*. However quantum systems of the same type are intrinsically identical.

> "If you were to study an individual quantum unit from a collection, you would find nothing to distinguish it from any other quantum unit of the same type. Nothing whatsoever. Upon regrouping the quantum units, you could not, even in principle, distinguish which was the unit you had been studying and which was another…
> Roger Penrose has likened this sameness to the images produced by a computer. Imagine the letter 't'. On the page you are viewing, the letter t appears many times. Every letter t is exactly like every other letter t. That is because on a computer, the letter t is produced by displaying a particular set of pixels on the screen. You could not, even in principle, tell one from the other because each is the identical image of a letter t. The formula for this image is buried in many layers of subroutines for displaying pixels, and the image does not change regardless of whether it is called upon to form part of the word 'mathematical' or 'marital'." [18]

Quantum Leaps

In classical physics whenever a change occurs it does so via a continuous shift from one state to another, however in quantum physics there are truly discontinuous leaps from one state to another. As if there were

> "discrete points and discrete instants with nothing in between, not even nothing…the mind reels to imagine space and time as disconnected, always seeking to understand what lies between two points or two instants which are said to be separate…In quantum mechanics, however, there is no transition at all. Electrons are in a low energy state on one observation, and in a higher energy state on the next; they spin one way at first, and in the opposite direction next. The processes proceed step-wise; but more than step-wise, there is no time or space in which the process exists in any intermediate state." [18]

Quantisation of Observables

In classical physics all quantities and concepts are continuous and are infinitely divisible, however in quantum physics this is not the case. It was found that certain calculations yielded infinities instead of sensible answers, however if it was assumed that space was quantised and there was a smallest possible distance, beyond which the concept of distance simply has no meaning, then these calculation yielded incredibly accurate results. The smallest distance is the Planck length, there are also Planck quantities for the smallest unit of time, frequency, energy, and so on. It would take an infinite amount of information to specify a quantity that was not quantised, hence quantisation makes sense from the perspective of system simulation.

> "It appears that there is no presently conceived computer architecture that would allow anything but such a discrete, digitized time and space, controlled by the computer's internal clock ticking one operation at a time. Accordingly, it seems that this lack of continuity, so bizarre and puzzling as a feature of our natural world, is an inherent characteristic of a computer simulation." [18]

Non-Locality

One of the most puzzling aspects of quantum systems, from a classical perspective, is non-locality.

> "[T]he essence of non locality is unmediated action-at-a-distance… Without benefit of mediation, a non-local interaction effortlessly flashes across the void…Even "flashes across the void" is a bit misleading, because 'flashing' implies movement, however quick, and 'across' implies distance traveled, however empty. In fact, non-locality simply does away with speed and distance, so that the

> cause and effect simply happen…There is no apparent transfer of energy at any speed, only an action here and a consequence there…The non-locality which appears to be a basic feature of our world also finds an analogy in the same metaphor of a computer simulation…In fact, the measured distance between any two pixels (dots) on the monitor's display turns out to be entirely irrelevant…The pixels may be as widely separated as you like, but the programming generating them is forever embedded in the computer's memory in such a way that – again speaking quite literally – the very concept of separation in space and time of the pixels has no meaning whatsoever for the stored information." [18]

Another analogy is a virtual reality flight simulator in which it may take hours to fly from one country to another in a virtual airplane, however the simulator itself does not operate within the context of that virtual space and time.

5.3 Consciousness

Naïve realism keeps us trapped within the veil of appearances, however via quantum mechanics modern science is beginning to penetrate through the veil of appearances and encounter the information processes that underlie the virtual reality that we think of as the physical universe.

Using the approach outlined in this book we are able to develop a detailed mathematical science that can comprehend the processual nature of existence and thereby understand consciousness and its place in reality.

Here we briefly examine the *hard problem* of consciousness and the general form of the *dynamical equations of individual consciousness*.

5.3.1 Hard Problem of Consciousness

> "The easy problems of consciousness include those of explaining the following phenomena:
>
> - the ability to discriminate, categorize, and react to environmental stimuli;
> - the integration of information by a cognitive system;
> - the reportability of mental states;
> - the ability of a system to access its own internal states;
> - the focus of attention;
> - the deliberate control of behavior;

- the difference between wakefulness and sleep.

There is no real issue about whether *these* phenomena can be explained scientifically. All of them are straightforwardly vulnerable to explanation in terms of computational or neural mechanisms...

If these phenomena were all there was to consciousness, then consciousness would not be much of a problem. Although we do not yet have anything close to a complete explanation of these phenomena, we have a clear idea of how we might go about explaining them.

The really hard problem of consciousness is the problem of *experience*. When we think and perceive, there is a whir of information-processing, but there is also a subjective aspect. As Nagel (1974) has put it, there is *something it is like* to be a conscious organism. This subjective aspect is experience. When we see, for example, we *experience* visual sensations: the felt quality of redness, the experience of dark and light, the quality of depth in a visual field...

Why is it that when our cognitive systems engage in visual and auditory information-processing, we have visual or auditory experience: the quality of deep blue, the sensation of middle C? How can we explain why there is something it is like to entertain a mental image, or to experience an emotion?

If any problem qualifies as *the* problem of consciousness, it is this one. In this central sense of "consciousness", an organism is conscious if there is something it is like to be that organism, and a mental state is conscious if there is something it is like to be in that state." [2]

Experiential Processes

We have touched upon the issue of 'experience' within the discussion so far by refering to the animating process operating 'within' a system as an *experiential process.* So far this term has been used to describe the fact that systems receive input information and are affected by it. However to what extent can they really be said to 'experience' it?

The systems paradigm presented in this book proposes that virtual systems do in fact 'experience' things in the full sense of that word. To comprehend this we must question what it is in our thinking that causes us to assume otherwise. A system can never observe another system's experiences, all it can do is observe that system's appearance and behaviour. Thus it is impossible to empirically prove that another system does or doesn't experience.

However it has been widely accepted that only certain systems can experience whilst others cannot. In many times and places it was assumed that only humans could experience, however now it is generally accepted that animals do as well. It is also accepted by many that in some sense plants also experience. All of this without a single shred of empirical evidence regarding the experiences of other systems; only observations of appearances (the content of our own experiences) and our interpretations of these observations within the context of some paradigm. Empirically speaking we can only ever verify our own experiential process and that of others is fundamentally unverifiable.

Rather than just accept a prevailing set of assumptions about which sytems have experiential processes and which are 'inanimate', let us try something different and see how it fits within the systems paradigm. For now, stop thinking of information processes in terms of inanimate physical processes, such as the functioning of a computer, and instead consider things from the perspective of neutral monism.

Neutral monism is a metaphysical philosophy that proposes that reality is neither purely physical nor purely mental, and neither is it composed of two fundamentally separate physical and mental aspects. Rather, there is a single aspect that is neither physical nor mental, which in some manner underlies both.

Consider the possibility that the underlying aspect can be conceptualised as an information process. This is in fact the crux of the systems paradigm that is being presented here.

This information process serves as a simulator that animates virtual systems. Through their input signals the virtual systems receive a view of a virtual reality, which from their perspective is indistinguishable from a 'physical' reality. It is a tangible dynamical context within which they find themselves. This explains the manner in which information processes can produce a flow of contents of perception that seems, in every respect, to portray a physical universe.

Regarding the mental aspect (felt inner experience) we cannot observe another's experiential process, however our own experiential process proves that systems can have such a thing as an experiential process. Furthermore, all of the easy problems of consciousness mentioned above can be explained in terms of information processing by sub-systems such as sensory organs, nervous systems, brains, neurons, synapses and so on. Thus awareness of felt experience is a subjective observation of the inner processes of many interacting sub-systems, which integrate and result in the experiential process of a super-system.

This is related to the way that interacting sub-systems produce the appearance of a single super-system via a meta system transition, which was discussed in sec. 2.5.1. Hence a meta system transition can be empirical; producing complex objects of perception ('physcial' objects), or it can be subjective; producing a complex experiential process (mind).

Thus our awareness of having a mind can be conceptualised as a subjective meta system transition by which the experiential processes of sub-systems appear to form into a single experiential process. Given that the sub-systems are animated by information processes, this explains the manner in which information processes can produce an experiential process that seems, in every respect, to be mental.

This line of enquiry suggests that the neutral monist aspect of reality can be conceptualised as an information process. Thus both physical and mental processes supervene on information processes, which are not "inanimate physical processes" but in fact produce genuine felt experience within virtual systems. It is for these reasons that the animating process, when embodied within a virtual system, is refered to as an experiential process.

Resolving the 'Hard' Problem

Within the context of the systems paradigm the "hard problem" is easily resolved. It is only difficult to resolve if one begins from assumptions about inert, inanimate systems. However in the systems paradigm inanimate systems do not exist. All systems are animate because they are animated into existence. All systems experience because the experiential process is what animates systems. Each point of existence experiences things from its perspective and responds. Thus the dynamical happening of existence is driven by experiential processes.

Related theories have been developed by many throughout history, in the form of panprotopsychism, panprotoexperientialism, process philosophy, neutral monism, Russellian monism, Type-F monism, strong inter-subjectivity, digital physics, computational metaphysics, Advaita Vedanta, Buddhism and so on. The theory presented in this book differs from these mainly in that it arises from a mathematical science of general systems; the resulting world-view however is compatible with these theories.

Inanimate or Animate Systems

It is difficult to imagine how inanimate systems can be governed by some mysterious 'laws' and what these laws might be such that inanimate systems can 'follow' them. It is also difficult to imagine how inanimate systems could interact at all let alone how their interactions could give rise to conscious experience - hence the difficulty in resolving the *hard* problem of consciousness within such a paradigm.

It is far more plausible that systems experience each other and respond, thereby changing state. It is only from a naïve realist perspective, which focuses solely on the appearances (content of experience), that one may postulate the existence of ‘inanimate’ systems and then formulate equations that describe the changing appearances. Then, being totally unaware of the underlying experiential processes that actually drive the system dynamics, it is proposed that the equations themselves are some kind of ‘law’ that the inanimate systems mysteriously ‘follow’.

In the context of the systems paradigm the “laws of physics” are descriptions of observable behaviour, whilst the causative principle is the animating process of the whole simulation, which gives rise to the experiential processes of all systems. The laws of physics provide insights into the structure of the system model that the simulator uses to animate the simulation, but they are not causative in themselves. Thus ‘experience’ is not something that must arise from the interactions of inanimate systems; rather, it is the driving force of all interactions between animate systems.

The naïve realist approach is analogous to someone observing a computer screen and formulating equations that describe the behaviour of the perceived objects, however these equations are descriptions only. They give insights into the underlying program, however the causative principle is the operation of the computer as it executes the program to animate the application, whose interface-objects the person is observing.

This example illustrates the degree to which naïve realism can lead us into false conclusions. For example, if a person is entirely unaware of the existence of computers and programs and they observed a computer screen assuming that the perceivable objects on the screen were fundamentally real objects (individual entities with self nature) they could only assume that descriptive equations are all that can be known about the situation. Thus when they pressed a button and a dialogue box came up, they would conclude that a fundamentally real button was pressed, which then resulted in the creation of a fundamentally real dialogue box. By realising that these objects are in fact virtual and that there is a program that defines them and a computer that animates them, the person can gain a much deeper understanding of the situation.

5.3.2 Dynamical Equations of Individual Consciousness

Within the classical context modelled by CSMN (sec.4.1) there exist individual classical systems with individual experiential processes, hence within this context SMN results in equations for individual experiential processes, which are in a sense *dynamical equations of individual consciousness*. These provide a mathematical framework for further research into consciousness.

Form of the Equations

As the universal equation ($V' = M \cdot V$) iterates there are many sub-equations encoded within this that are also iterated. Each of these sub-processes represents the animating process of a classical system. The universal equation can be examined to identify the individual sub-equations that represent the structure and behaviour of an individual experiential process. Or conversely, the individual equations can be defined to build up the form of the universal equation.

These experiential processes can be thought of as streams of individual consciousness. These streams of consciousness perceive, experience, respond and evolve as the systems interact with each other. Thus the universal equation is the dynamical equation of universal consciousness and the sub-equations are dynamical equations of individual consciousness.

Here we first illustrate the form of the equations, then explain what the components represent in terms of consciousness and how the components evolve.

Recall that in CSMN the matrix M has elements $m_{ij} = I_{ij} \cdot O_{ij}$. Note that these are operators so $I_{ij} \cdot O_{ij} \equiv I_{ij}(O_{ij}())$. When the matrix is multiplied with a state vector using *generalised* matrix multiplication (sec. 4.1) we get (for two systems)

$$M \cdot V = \begin{pmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} \tag{5.1}$$

$$= \begin{pmatrix} R_0\Big(P_{00}(m_{00}, v_0), P_{01}(m_{01}, v_1)\Big) \\ R_1\Big(P_{10}(m_{10}, v_0), P_{11}(m_{11}, v_1)\Big) \end{pmatrix}$$

Here we define the pairwise function $P()$ as multiplication but leave $R()$ in general form. Thus in this case

$$\begin{pmatrix} R_0\Big(P_{00}(m_{00}, v_0), P_{01}(m_{01}, v_1)\Big) \\ R_1\Big(P_{10}(m_{10}, v_0), P_{11}(m_{11}, v_1)\Big) \end{pmatrix} = \begin{pmatrix} R_0(m_{00} \cdot v_0), (m_{01} \cdot v_1) \\ R_1(m_{10} \cdot v_0), (m_{11} \cdot v_1) \end{pmatrix}$$

$$= \begin{pmatrix} R_0\Big((I_{00} \cdot O_{00} \cdot v_0), (I_{01} \cdot O_{01} \cdot v_1)\Big) \\ R_1\Big((I_{10} \cdot O_{10} \cdot v_0), (I_{11} \cdot O_{11} \cdot v_1)\Big) \end{pmatrix} \tag{5.2}$$

Note that the operator equation $I_{ij} \cdot O_{ij} \cdot v_j$ is equivalent to the function equation $I_{ij}(O_{ij}(v_j))$.

Components of the Equations

- v_j represents system j's internal state.
- O_{ij} represents system j's output filter that expresses its internal state to system i.
- $(O_{ij} \cdot v_j)$ is the raw appearance of system j from system i's perspective.
- I_{ij} represents system i's input filter that interprets the raw appearance of system j.
- $(I_{ij} \cdot O_{ij} \cdot v_j)$ is system i's experience of system j.
- The set $F_i = \{I_{ij} \cdot O_{ij} \cdot v_j \mid \forall j\}$ is system i's field of experience, which includes experiences of the raw appearance of every perceivable system including itself.
- $R_i(F_i)$ is a row operation that integrates all of system i's experiences and responds with an new internal state for system i.

Equations of Evolving Consciousness

For each individual system (indexed by i) there are several equations that describe its perceptions, experiences and responses over time.

All of the components can vary from iteration to iteration, thus a system's experiential process can evolve due to its stream of experiences, responses and expressions. The state of a system's input and output filters and its response function can be thought of as its *attitude*. Thus its attitude can change due to its interactions with systems.

The equations represents a cognitive feedback loop. Thus 'how' a system experiences influences 'what' it experiences, and 'what' a system experiences influences 'how' it experiences.

Each component below corresponds either to the current value (unprimed) or the new value (primed).

For system i:

Changing state (state of being):

$$v'_i = R_i(F_i) \text{ where } F_i = \{I_{ij} \cdot O_{ij} \cdot v_j \mid \forall j\} \tag{5.3}$$

Changing response mapping (state of mind):

$$R'_i = \Omega(R_i, F_i) \tag{5.4}$$

For each j:

Changing output interface (expression):

$$O'_{ij} = \Phi(O_{ij}, v_j) \tag{5.5}$$

Changing input interface (perception):

$$I'_{ij} = \Theta(I_{ij}, O_{ij} \cdot v_j) \tag{5.6}$$

These describe the detailed processes that implement a system's evolving attitude.

This determines where it directs its attention, how it interprets what it perceives, how these experiences impact upon the perceiving system's internal state and how its attitude changes due to its experiences, responses and expressions.

The exact form of the evolution functions $\Omega(), \Phi()$ and $\Theta()$ determine how the system's state of mind, way of expressing and way of perceiving change. Each function changes an aspect of the system's attitude based on the current state of that aspect and the current information flowing through it.

We have examined a single iteration of these equations whereby the current internal state is filtered and processed to produce a new internal state. However with each iteration the internal state is further filtered and processed, thus with successive iterations the equations are recursively nested.

By giving each component a temporal index $[t]$ we can examine a second iteration and from there infer the form of successive iterations. From the above example of two systems we see that after one iteration the new internal state of system 0 is

$$v_0[1] = R_0[0]\Big((I_{00}[0] \cdot O_{00}[0] \cdot v_0[0]), (I_{01}[0] \cdot O_{01}[0] \cdot v_1[0])\Big) \tag{5.7}$$

After another iteration the new internal state of system 0 is

$$v_0[2] = R_0[1]\Big((I_{00}[1] \cdot O_{00}[1] \cdot v_0[1]), (I_{01}[1] \cdot O_{01}[1] \cdot v_1[1])\Big) \tag{5.8}$$

Where

$$R_0[1] = \Omega(R_0[0], F_0[0]) \tag{5.9}$$

$$I_{00}[1] = \Theta(I_{00}[0], O_{00}[0] \cdot v_0[0]) \tag{5.10}$$

$$O_{00}[1] = \Phi(O_{00}[0], v_0[0]]) \tag{5.11}$$

$$v_0[1] = R_0[0]\Big((I_{00}[0] \cdot O_{00}[0] \cdot v_0[0]), (I_{01}[0] \cdot O_{01}[0] \cdot v_1[0])\Big) \tag{5.12}$$

$$I_{01}[1] = \Theta(I_{01}[0], O_{01}[0] \cdot v_1[0]) \tag{5.13}$$

$$O_{01}[1] = \Phi(O_{01}[0], v_1[0]]) \tag{5.14}$$

$$v_1[1] = R_1[0]\Big((I_{10}[0] \cdot O_{10}[0] \cdot v_0[0]), (I_{11}[0] \cdot O_{11}[0] \cdot v_1[0])\Big) \tag{5.15}$$

Substituting these into the equation for $v_0[2]$ (5.8) we get an equation in terms of initial conditions. Thus for a particular iteration $[n]$ the equation is first expressed in terms of the previous iteration $[n-1]$, then via substitution we can express it in terms of any previous iteration, right back to the initial state [0]. This can be applied to the entire equation or to particular branches of the recursively nested structure.

In a typical complex system there may be millions of interacting systems and millions of iterations, hence these will be extremely complex equations. Nevertheless they express the current state of a system in terms of its entire stream of experiences and the consequent evolvution of its attitude.

These overall dynamical equations of individual consciousness require a great deal of further development in order to be able to model anything that would be recognisable to ourselves as conscious systems, however they provide a way of working towards that. A way to do so would be to identify evolution functions for simple systems and build complex networks of those systems. This would effectively produce complex systems with complex evolution functions.

This set of equations also make it explicit that SMN is an inherently panpsychist approach, where every system has a stream of consciousness and these streams can be modelled and analysed using CSMN.

5.4 System Science

From the discussion so far we can provide a concise definition of a *system*. A system can be described as *a dynamic pattern of information that structures the flow of information*. Recall that information here is defined as discernible difference. A system is metaphorically like a whirlpool in a stream, which is a pattern of water that structures the flow of water. However a system is not an entity in space and time, it consists of structured information flowing through the simulator. Most of a system is imperceptible and it is only the appearances of systems that can be discerned within the virtual reality.

From this definition there are several rammifications:

Inter-Connectedness

Systems seem to be separate individual entities when their appearances are observed by a system. However aside from these appearances a system is an intrinsic and inseperable part of a unified and universal animating process. It is not a mereological 'part' in the sense that the animating process is "made of" systems, however it is a part in the sense that it IS the animating process but not the whole of it.

Global Coherence

The unified foundation and inter-connectedness result in global coherence between all systems so all interactions are part of a unified process and are not purely local interactions between intrinsically separate systems. Every event is a 'movement' of the whole.

No Fundamental Existence to Objects

Objects are the appearances of systems and a system has no fundamental self nature, but is instead an emergent property of the functioning of the simulator.

Two Existential Contexts

These are not separate but are different conceptions of the same context. The unified context can be conceptualised as either a computational or virtual context, a processual or experiential context, an absolute or relative context. These pairs of conceptualisations depend on whether one focuses on information processes or the information content within these processes. I.e. they depend on whether one observes experiential processes (subjectivism) or the content of experience (empiricism).

Subjectivity

All observable phenomena are the *content* of an experiential process. Thus naïve realism, which ascribes fundamental existence to appearances is a flawed position. Nothing can be observed *objectively* because all observations are subjective experiences. Objectivity can only be inferred through communication and repeated subjective observation.

Virtualware

In traditional VR technology the appearances of complex systems are animated by computational processes (software). Thus a virtual system cannot disect other systems to discern how they operate nor can it decompose systems into sub-systems that can then be recombined into a new system. All of the details of a system are specified within the programming which is inaccessible to virtual systems.

However in an SMN simulation every complex system consists of sub-systems, each of which has an observable appearance. Thus the causal structure and dynamics of systems is not hidden within the programming and any system can be dissected or decomposed and new systems can be formed. In a systemic virtual world the 'software' (SMN simulator) simply provides potential existence and everything else if defined in 'virtualware', which can be examined and manipulated by virtual systems.

Thus it is possible for virtual systems to conduct science and technology within an SMN generated virtual reality. Furthermore, a mature system science would provide for the development of science and technology to levels of sophistication that would be impossible to reach otherwise.

Simulation - Analogy or Literal Fact?

Given the common definitions of terms, the idea that the physical universe is a virtual reality simulation being animated by a simulator, which is a program running on a computer is just a useful analogy and should not be taken literally. However given the general definitions of information as discernible difference, computation as the coherent transformation of information and the simulator as a self-referential computational process that manifests both experiential processes and the content of experience. In this general sense it is a literal fact that the physical universe is only known via the content of experience, that all systems are animated by experiential processes and that the world-of-appearances (experiential context) is not self-existent but is instead an emergent phenomenon arising from underlying information processes.

Therefore, in the context of these general definitions and particularly in light of quantum physics the ideas of simulator, simulation, virtual systems and virtual reality represent a literal description of the nature of reality. Thus the general theory of virtual reality and the systems paradigm imply that the physical universe is virtual.

Web Ontologies

Current upper level web ontologies[1] begin with abstract concepts but then define spatio-temporal contexts and objects as if these have fundamental existence. Therefore they are naïve realist. To be realistic they need to provide more detail at the deeper levels.

A systems ontology begins with information processes and information spaces. It then defines an SMN-like simulator that produces a simulation. The dynamic patterns of information can then be conceptualised as virtual systems. These systems have experiences and due to naïve realism the experiential contexts are interpreted as an external world.

It is within these subjective world-experiences that concepts such as spatio-temporal contexts and objects arise. From this point on a systems ontology would be the same as current web ontologies however the inferences drawn from it would differ due to its system theoretic rather than naïve realist conceptual foundation.

For instance, 'matter' could not be defined as a fundamental substance in which physical properties inhere, rather it is a metaphor that systems use to comprehend their tangible experiences of system dynamics.

Ludwig von Bertalanffy

Finally a few comments regarding system science from one of its founders, Ludwig von Bertalanffy:

> "Our civilization seems to be suffering a second curse of Babel: Just as the human race builds a tower of knowledge that reaches to the heavens, we are stricken by a malady in which we find ourselves attempting to communicate with each other in countless tongues of scientific specialization... the only goal of science appeared to be analytical, i.e., the splitting up of reality into ever smaller units and

[1] A web ontology is a digitally specified ontology that defines all concepts and relations believed to exist in some context. An upper level ontology specifies the fundamental context within which all other ontologies are situated. See [20, 21, 22, 23].

> the isolation of individual causal trains... We may state as characteristic of modern science that this scheme of isolable units acting in one-way causality has proven to be insufficient. Hence the appearance, in all fields of science, of notions like wholeness, holistic, organismic, gestalt, etc., which all signify that, in the last resort, we must think in terms of systems of elements in mutual interaction." [24]

> "General Systems theory should be an important means of instigating the transfer of principles from one field to another (so that it would) no longer be necessary to duplicate the discovery of the same principles in different fields." [24]

> "There is this hope, I cannot promise you whether or when it will be realized - that the mechanistic paradigm, with all its implications in science as well as in society and our own private life, will be replaced by an organismic or systems paradigm that will offer new pathways for our presently schizophrenic and self-destructive civilization." [24]

5.5 Naïve Realism

The most ubiquitous false belief system, that has beguiled humanity throughout history has, in recent times, been clearly revealed to be fundamentally flawed. A paradigm shift is characterised as much by the assumptions and pre-conceptions that we discard as by the realisations that we take up. The deeper the assumptions are the more profound are the realisations and the more revolutionary is the paradigm shift.

Naïve realism is a chronic habit that is deeply conditioned, not just culturally conditioned and biologically conditioned. It is an inherent precondition for a system to experience a virtual reality, thus all virtual systems are naïve realist by nature; they all accept their input information without question. It takes a sentient being with a great level of self-awareness and intelligence to discern its naïve realist tendencies and even then these remain as an entrenched habit. It takes great persistence to overcome this habit, because all of ones interpretations and attachments reinforce the habit. A complete and total paradigm shift is required and there is no deeper paradigm shift that a system can undergo. By overcoming naïve realism they realise that they are not an object *in* a world, they are the universal animating process itself having a virtual world-experience within which it experiences itself as an individual entity, and so too are all systems.

Once that realisation overcomes the entrenched naïve realist legacy of the past and permeates our minds and the collective culture, radical changes will naturally arise throughout all aspects of our lives and civilisation as a whole. Then many endemic dysfunctions will evaporate and new possibilities will arise that were previously inconceivable.

Bibliography

[1] T. S. Kuhn, *The Structure of Scientific Revolutions*. University Of Chicago Press, 1962. p103.

[2] D. J. Chalmers, "Facing up to the problem of consciousness," *Journal of Consciousness Studies*, vol. 2, no. 3, pp. 200–19, 1995. http://consc.net/papers/facing.html.

[3] W. A. Stein, *Sage Open Source Mathematics Software (Version 3.2.3)*, 2009. http://www.sagemath.org/.

[4] Wikipedia, *Norm(mathematics)*. http://en.wikipedia.org/wiki/Norm_(mathematics).

[5] M. Dubson, S. McKagan, and C. Wieman, *Stern-Gerlach Experiment Simulation*. http://phet.colorado.edu/sims/stern-gerlach/stern-gerlach_en.html.

[6] E. C. Kemble, *The Fundamental Principles of Quantum Mechanics*. McGraw Hill (reprinted by Dover), 1937.

[7] Wikipedia, *Naïve realism*. http://en.wikipedia.org/w/index.php?title=Naive_realism&oldid=276101166.

[8] J. D. Jackson, *Mathematics for Quantum Mechanics*. Dover, 2006.

[9] T. F. Jordan, *Linear Operators for Quantum Mechanics*. Dover, 2006.

[10] T. F. Jordan, *Quantum Mechanics in Simple Matrix Form*. Dover, 2005.

[11] P. Ball, "Physicists bid farewell to reality? quantum mechanics just got even stranger.," *Nature*, 2007. http://www.nature.com/news/2007/070416/full/news070416-9.html this article has been removed by Nature but excerpts can be read at http://www.freerepublic.com/focus/f-news/1820354/posts.

[12] B. Rosenblum and F. Kuttner, *Quantum Enigma: Physics Encounters Consciousness*. Oxford University Press, 2006. pg 112, http://books.google.com.au/books?id=hKrvv35-gcMC.

[13] C. P. Williams and S. H. Clearwater, *Explorations in Quantum Computing*. Springer - Telos, 1998.

[14] E. W. Weisstein, *Kronecker (Direct) Product*. MathWorld – A Wolfram Web Resource, http://mathworld.wolfram.com/KroneckerProduct.html.

[15] J. Ringland, *SMN Software Methodology*. (discussion and source code) http://www.anandavala.info/SMN/technology.html.

[16] J. Ringland, *SMN System Modelling Examples*. http://www.anandavala.info/TASTMOTNOR/SMN%20Code.html.

[17] J. Ringland, *Finite Discrete Information Systems*. Note: that which is called 'quantum' here is actually PSMN. http://www.anandavala.info/TASTMOTNOR/Finite%20Discrete%20Information%20Systems.html.

[18] R. Rhodes, *A Cybernetic Interpretation of Quantum Mechanics*, 2001. http://www.bottomlayer.com/bottom/argument/Argument4.html.

[19] R. Penrose, *Quantum Concepts in Space and Time*, ch. Gravity and State Vector Reduction. Oxford: Clarendon Press, 1986.

[20] Wikipedia, *Web Ontology Language (OWL)*. http://en.wikipedia.org/wiki/Web_Ontology_Language.

[21] Laboratory for Applied Ontology, Institute of Cognitive Science and Technology, Italian National Research Council, *Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)*. http://www.loa-cnr.it/DOLCE.html.

[22] IEEE, *Suggested Upper Merged Ontology (SUMO)*. http://www.ontologyportal.org/.

[23] Institute for Formal Ontology and Medical Information Science (IFOMIS), Saarland University, *Basic Formal Ontology (BFO)*. http://www.ifomis.org/bfo.

[24] T. Mandel, ed., *The General System*. Primer Group, International Society for the Systems Sciences, 1997.

Index