COMPUTATIONAL FORECASTING

Can we use a computer to predict the future?

Abstract

In this dissertation, I attempt to answer the question of whether it will be possible to predict exactly the future using a computer. The goal is achieved by exploring relevant topics in mathematical modelling, theoretical physics, computer science, and engineering. The research is then discussed, before concluding that exact prediction is most likely impossible, particularly due to the impossibility of precisely measuring the state of the universe. However, the essay also concludes that computers can still be used to create useful estimates using empirical models. In the conclusion, some of the ethical issues related to the topic are mentioned, including ideas related to determinism, free will, and impacts of the technology on future society. Finally, the paper reviews some of the shortcomings of the dissertation and proposes opportunities for any further research.

Miles Watson

December 2020

Table of Contents

1 Introduction

Since programmable computers were first created, scientists have used them to make predictions about the world around us with ever increasing accuracy. From predicting ballistic trajectories to evaluating the feasibility of thermonuclear weapons, modelling everything from protein folding to the motion of planets, computer simulation has played a pivotal role in recent world history.

Throughout this dissertation, I attempt to push the theoretical limits of computational forecasting, with the aim of exploring the concept of a perfect universe simulation using a very powerful supercomputer. I do this by splitting the research into three chapters.

In the first chapter, I introduce the concept of a mathematical model, and explore the properties of complex systems – particularly looking at ideas of chaos and emergence. I also quickly review the difference between mechanistic and empirical modelling, as well as techniques that can be used to provide predictions from stochastic (probabilistic) models.

In the second chapter, I review the theoretical groundwork that Turing laid for digital computers and provide a summary of the basic challenges of computer engineering – this includes the challenges associated with building massive, galactic supercomputers known as "Matrioshka brains". In addition, I talk about alternative computing technologies, such as experimental transistor types and quantum computers, as well as a brief overview of machine learning.

In the third chapter, I write about current progress towards finding a theory of everything, in the hopes of creating a precise mathematical model of the universe. The chapter also looks at the fundamental limits to measurement and information storage, providing some relevant calculations for perspective.

In the discussion section, I lay out what I believe to be the criteria that must be satisfied in order for computers to be able to perfectly predict the future. Next, I evaluate each one by combining research in the previous chapters with my own ideas. Afterwards, I provide some ways in which a hypothetical simulation could be optimised, at the expense of inaccuracy.

I use the conclusion section to review my findings, briefly touching on some of the philosophical and ethical issues related to them, before discussing some of the limitations of the dissertation's methodology. Finally, I evaluate the success of the project, and talk about some of the lessons I've learned from the process of tackling such a project.

2 Mathematical Modelling

2.1 Introduction to Modelling

Throughout history, many attempts have been made to replicate the way our universe functions in the language of mathematics. In their paper on mechanical modelling[1], Marion and Lawson briefly describe the concept and advantages of mathematical modelling:

- *"Models describe our beliefs about how the world functions. In mathematical modelling, we translate those beliefs into the language of mathematics. This has many advantages:*
  1. *Mathematics is a very precise language. This helps us to formulate ideas and identify underlying assumptions.*
  2. *Mathematics is a concise language, with well-defined rules for manipulations.*
  3. *All the results that mathematicians have proved over hundreds of years are at our disposal.*
  4. *Computers can be used to perform numerical calculations."*

It is therefore clear that in order to describe the behaviour of a system precisely, concisely, and usefully, the language of mathematics is critical.

One of the most famous examples of describing a model using mathematics is Newton's laws of classical motion. In his book "Philosophiae Naturalis Principia Mathematica"[2], Newton outlines his findings on the mathematical nature of motion. In particular, the second law describes the relation between force and "motion".

- *"If any force generates a motion, a double force will generate double the motion, a triple force triple the motion…"*

In this case, the mathematical model describes the relationship between force and "motion" as directly proportional. His definition of "motion" can be taken to mean change in momentum – in common usage the equation is rearranged to give:

$$F = \frac{d\boldsymbol{p}}{dt} = \frac{d(m\boldsymbol{v})}{dt} = m\frac{d\boldsymbol{v}}{dt} = m\boldsymbol{a}$$

This is arguably one of the simplest examples of a useful mathematical model – it is used throughout almost every aspect of mechanics.

---

[1] (Marion, et al., 2008) – A paper that reviews the process of building, studying, testing, and using mathematical models. There are only a few citations, and I did notice one spelling error. However, the information is reasonable and concise, and Marion is linked with a reputable research organisation (BioSS).

[2] (Newton, 1687) – A published copy of Andrew Motte's translation of Principia Mathematica. Some minor details may have been lost or misconstrued in the translation, but the edition is well known and respected. The extract refers to Newton's three laws of motion – objects remain at a constant velocity unless acted upon, $F = ma$, and that for every force there is an equal and opposite reactive force.

2.2 Systems

Although rules such as $F = ma$ are useful, our universe is considerably more complex than a simple relationship between three variables. Physicists tend to describe the universe as a collection of objects that interact with each other – usually this results in them creating a mathematical system that mimics the behaviour of our universe. A system is defined by the Merriam-Webster online dictionary[3] to be:

- *"a regularly interacting or interdependent group of items forming a unified whole"*

Systems are often thought of as collections of elements, each with properties that change over time. The changes of each property are governed by a set of rules which determine the way the system evolves.

One way to represent systems is by using graphs – collections of vertices (elements) linked with edges (interactions). Assuming each element in the system has an interaction with every element other than itself [*Figure 1*], the total number of interactions (edges, E) can be written in terms of the number of elements (vertices, V):

$$E = \frac{V(V-1)}{2} = \frac{1}{2}V^2 - \frac{1}{2}V$$

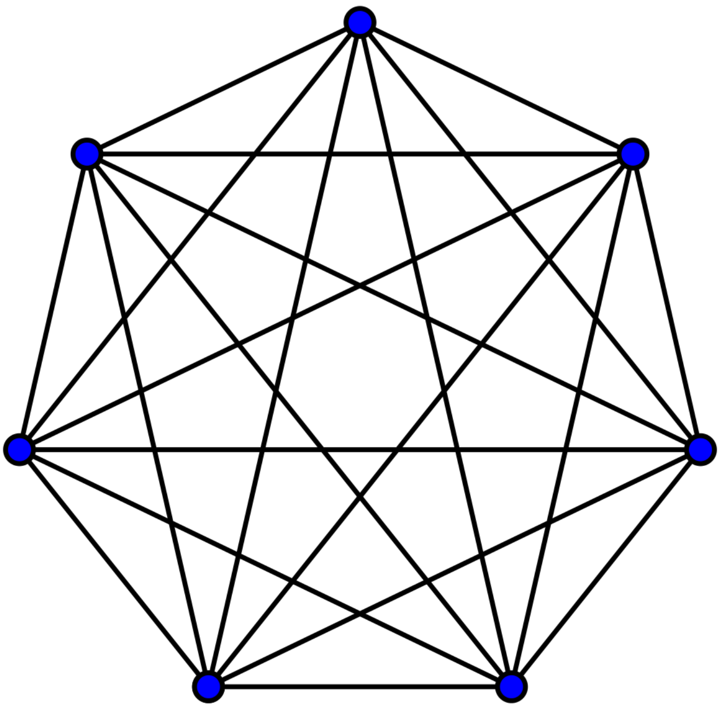


Figure 1 – The complete graph on 7 vertices (Benbennick)

There are $\frac{6\times 7}{2} = 21$ *edges in total.*

[3] (Merriam-Webster) - Definition of "System". Reputable dictionary, definition is reasonable and agrees with other literature, although there are no sources listed for the information.

As the number of vertices increases the squared term quickly outweighs the linear term:

$$\lim_{V\to\infty} E = \frac{1}{2}V^2$$

$$\lim_{V\to\infty} E \propto V^2$$

From this it can be seen that, given a sufficiently large number of items, doubling the number of modelled objects quadruples the number of interactions.

2.3 Mechanistic Models of Systems

There are two main types of models – mechanistic and empirical models. Mechanistic modelling (also known as intrinsic modelling) involves describing as many underlying features of the system as possible, whilst empirical modelling (also known as extrinsic modelling) involves modelling the relationships between measured data items.

To create a mechanistic model, there must be an understanding of the features of the model. This makes it ideal for situations where the elements of the system (and their interaction) can be easily observed. For example, take a system comprised of N massive bodies (the elements) which are influenced by gravitational forces (the interactions). Using Newton's law for universal gravitation (and therefore assuming absolute space and time), the system can be described as follows:

- Each element i has 3 properties – a mass scalar m, a position vector s, and a velocity vector v.
- Behaviour can be given by:

$$\frac{dm_i}{dt} = 0$$

$$\frac{d\boldsymbol{s}_i}{dt} = \boldsymbol{v}_i$$

$$\frac{d\boldsymbol{v}_i}{dt} = \sum_{n=1}^{N} G\frac{m_i(\boldsymbol{S}_n - \boldsymbol{S}_i)}{|\boldsymbol{S}_n - \boldsymbol{S}_i|^3}[n \neq i]$$

The differential equations of $N = 2$ (the two-body problem) were analysed by Johannes Kepler in 1609, and then solved by Isaac Newton in 1687. The equations derived from the system can be used to model easily and predict the interactions of two bodies. However, this is not the case for the differential equations of $N \geq 3$, none of which (despite the efforts of many famous scientists and mathematicians including Newton and Poincaré) we have been able to solve[4].

[4] (Wolfram, 2002) - A brief look into the history of the three-body problem. The author is an inaugural fellow of the American Mathematical Society - he is known for his work in computer science and theoretical physics. Furthermore, he spearheaded the development of Mathematica and the Wolfram Alpha answer engine. There are no citations for this section, but both the book and publisher are reputable enough to be trusted.

2.4 Chaos

Chaos theory is a discipline of mathematics and science that doesn't focus on that which can be predicted – rather, it focuses on that which can't.

Although the idea of chaos has roots back to Poincaré, the subject has only recently become relevant due to the invention of computers and computer simulation. The discovery of modern chaos theory is often credited to Edward Lorentz, who worked on simulating weather patterns on a digital computer. He noticed that slightly tweaking the input values to the simulation resulted in greatly different outcomes[5]. The idea became popularised as the "Butterfly Effect" – the flap of a butterfly's wings can lead to cascading events culminating in a hurricane on the other side of the world.

The definition of chaos was further refined by Robert Devaney in his paper on chaotic dynamical systems. He gave a concise definition for chaos[6]:

- Let S be a set. The continuous mapping $f: S \to S$ is said to be chaotic on S if:
    1. f has sensitive dependence on initial conditions
    2. f is topologically transitive
    3. f has periodic points that are dense in S

These points can be broken down to gain a deeper understanding of the behaviour and properties of chaotic systems [*Appendix A*]. To summarise: chaotic maps have three qualities: unpredictability, indecomposability, and an element of regularity. They are unpredictable because they are sensitive to initial conditions, they cannot be split into two subsystems because of topological transitivity, and there is an element of regularity in dense periodic points.

The element of regularity is perhaps the most surprising of the characteristics. The idea was first demonstrated when Lorenz tried to create the simplest version of his weather system that exhibited chaotic behaviour. When σ, ρ, and β are constants, the Lorenz system is described as follows:

$$\frac{dx}{dt} = \sigma(y - x)$$

$$\frac{dy}{dt} = x(\rho - z) - y$$

$$\frac{dz}{dt} = xy - \beta z$$

[5] (Deterministic Nonperiodic Flow, 1963) – The original paper by Lorenz, includes the equations and diagrams for the Lorenz attractor.

[6] (Devaney, 2003) – Provides a detailed definition of chaos. Robert Devaney is an award-winning Professor at Boston University, and an inaugural member of the American Mathematical Society (and so I consider the author reliable). The book provides further reading but no citations, which limits the reliability of the source.

For certain parameters, the dense periodic points manifest themselves as interesting patterns [*Figure 2*].

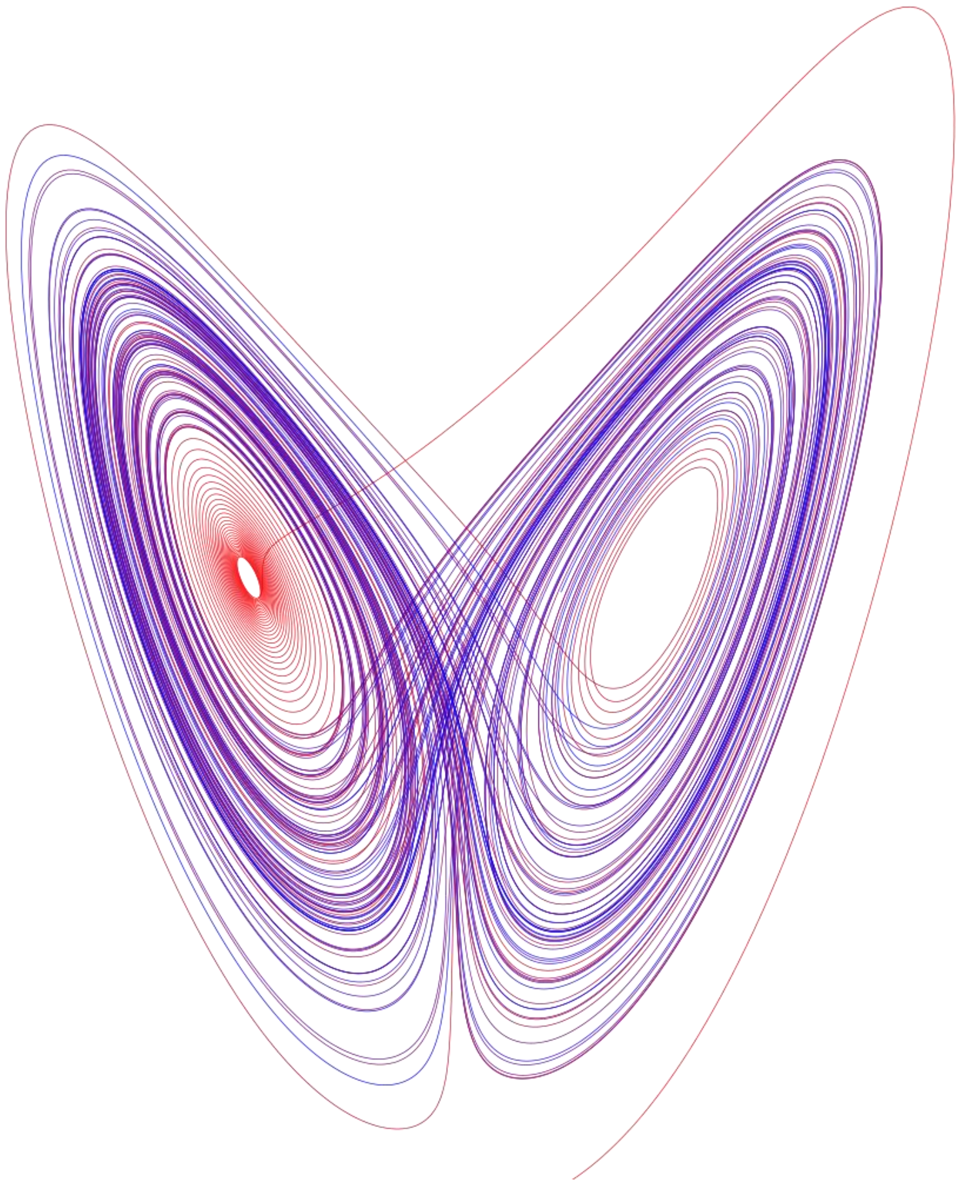


Figure 2 – A solution in the Lorenz attractor (Dschwen)

Despite the model being chaotic, its dense periodic points manifest themselves as a regular pattern.

2.5 Complex Systems

If a simple chaotic model can have emergent regular structures, it stands to reason that the same might be true for chaotic systems with large numbers of elements. In the final sections of their paper on complex systems[7], Ladyman et al. define a complex system to be:

- *"an ensemble of many elements which are interacting in a disordered way, resulting in robust organisation and memory"*

They concluded that, in order for a system to be considered complex, it must have:

1. Many elements
2. Interactions between elements
3. Disorder at the micro level
4. Robust order at the macro level
5. Memory (a corollary of robust order)

For example, cells in the human body can be seen as elements of a complex system. There are many cells, which interact using electrical, physical, and chemical signals. There is disorder at the micro level, yet there is an emergence of robust order at the macro level (the human body is consistent and not fragile).

From this, it is easy to see how links can be drawn between our universe and complex systems – in the standard model, our universe is comprised of many vertices (particles) connected by edges (fundamental interactions). There is disorder at the micro level due to high uncertainty and quantum phenomena, but this disorder forms robust pattern and structure.

2.6 Stochastic Modelling and Ensemble Forecasting

There often instances where deterministic systems are not suitable to describe useful phenomena. For example, when modelling a coin flip, it may be possible to setup a system of equations which simulate all the factors involved in deciding the outcome. However, it is much simpler to describe a coin flip stochastically: it has a 50% chance of being heads and a 50% chance of being tails. Each time the model is used, a different value for the coin result is generated. By repeatedly sampling from more complex models, results can be analysed to provide insight into complex deterministic problems. The collection of techniques that use this concept are known as "Monte Carlo simulations".

[7] (What is a Complex System?, 2013) – A discussion on the definition and properties of complex systems. Both Ladyman and Lambert work in Dept. of Philosophy at Bristol University, whilst Wiesner works in the Department of Mathematics at the same university. The paper is well cited, the authors are reputable, and the paper is linked to a well-known university – I therefore believe the paper to be a reliable source.

The randomness doesn't have to be binary and can instead be from any probability distribution. One common technique was developed by Edward Epstein – in his 1969 paper on weather forecasting he stated[8]:

- *"Stochastic dynamic prediction assumes the laws governing atmospheric behaviour are entirely deterministic, but seeks solutions corresponding to probabilistic statements of the initial conditions, thus recognising the impossibility of exact or sufficiently dense observations."*

In the paper he explored the idea of randomly varying atmospheric readings to simulate other possible initial states (using calculated uncertainty from the actual reading). By repeatedly doing this, a probability distribution of many possible outcomes can be calculated. The technique is known as "ensemble forecasting", and makes the forecast much more versatile, as there is less chance of an anomalous outcome (due to the results being averaged).

[8] (Stochastic Dynamic Prediction, 1969) – A paper by a professor in meteorology. Both the paper and the author are well known, with the former being well cited and having references throughout.

3 Computer Science and Engineering

3.1 Turing Machines and Computable Numbers

In his 1936 paper, Turing established the concept of an “automatic machine” [9]. The described machine involved a movable processor “head” that could read from and write to an infinite “tape” stamped with “symbols”. Contained within the processor head is some finite amount of memory, the state of which is combined with the most recently scanned symbol to determine a course of action according to a state-transition table (the actual workings of the model are more specific, see *Appendix B*). When an automatic machine consists of only two symbols (0 and 1), he called it a “computing machine”.

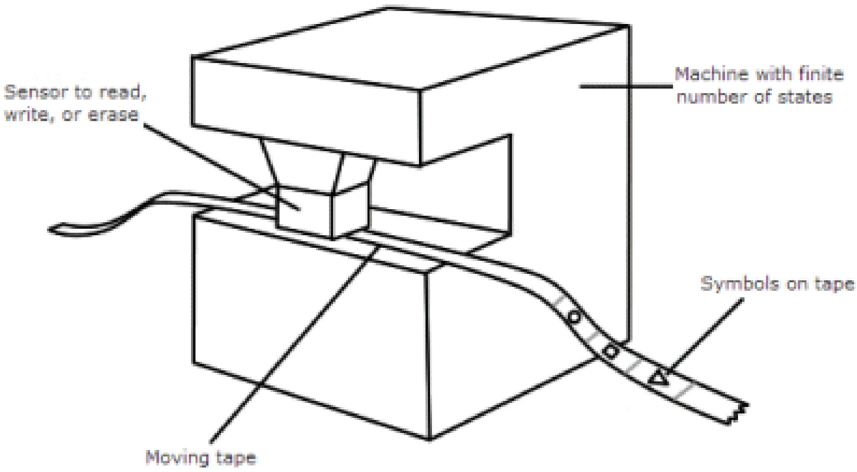


Figure 3 - A diagram explaining the concept of a Turing machine (CSL4U)

After introducing the concept of a computing machine, Turing establishes the concept of a “computable number”. A computable number is any real number that can be computed to within any desired precision by a finite, terminating program. For example, although π is irrational, it is computable as there is a finite algorithm that allows for its computation to any desired number of significant figures[10]. The existence of computable numbers implies the existence of non-computable numbers – numbers that Turing machines can never display or calculate with. In fact, it can be proved that almost all[11] real numbers are non-computable.

3.2 Computer Engineering

Although modern computers retain similar functionality to Turing machines, they function quite differently. Computer hardware can be incredibly complex, but at a high level there are three main

[9] (On Computable Numbers, with an Application to the Entscheidungsproblem, 1937) – Turing’s original paper where he lays the foundation for “computing machines”. In the paper, he describes the concept of an “automatic” machine and establishes the concept of a computable number.
[10] (The Computation of π to 29,360,000 Decimal Digits Using Borweins' Quartically Convergent Algorithm, 1988) – A paper by an award-winning mathematician and computer science about the process of calculating π on a NASA Cray-2 supercomputer.
[11] “Almost all” here should be taken to mean “all elements of an uncountable set except countably many”.

components – a Central Processing Unit (CPU), primary memory (RAM, fast but low capacity), and slow secondary storage (e.g. a hard disk drive, slow but high capacity). The processor is a carefully designed arrangement of electronic switches (transistors) that sequentially executes instructions stored in RAM. If the processor needs to store more data than there is space for in RAM, parts of the RAM are swapped into secondary storage when they are not needed, and then swapped back to RAM when they are needed.

There are two main properties of a CPU that affect its performance – clock speed and the number of cores. Clock speed is the number of instruction-cycles that the CPU can execute per second, measured in Hz. A higher clock speed means that a CPU can process more instructions per second – this leads to better simulation performance, meaning a running simulation can be bigger, more complex, and/or quicker. However, physical limitations constrain how high the clock speed of a processor can rise (e.g. before the high temperature causes damage to the components). Furthermore, a phenomenon known as "quantum tunnelling" restricts the size of transistors within the CPU[12], which may limit the clock speed that electronic digital computers will ever be able to reach.

To circumvent this issue, chip designers started putting multiple processors (called cores) into a single CPU. Cores can share memory, but each one runs a different program at the same time. This means that, theoretically, a dual core processor should have twice the performance as a comparable single core processor – a concept called "parallel computing". Multiple independent computers (nodes) can also be connected to form one giant "processor" – a cluster computer. This usually involves using network hardware to allow the nodes to share resources and results.

Unfortunately, there are even limits to the number of compute units within a cluster computer. In his paper titled "Matrioshka Brains", Bradbury explores the theoretical limits to how powerful computers will ever be able to get based on the idea of star, galaxy, and universe sized supercomputers[13]. He concluded that the fundamental limits to how effectively a Matrioshka brain could perform were bandwidth and latency.

Bandwidth is important as it dictates the amount of information which can be transferred from one node to another in a given time frame. This is important in many situations, most notably if a large database is distributed across a cluster computer: the transfer speed could bottleneck the computation if the demand is high enough.

Latency, on the other hand, determines the delay between requesting a resource and receiving it. Bradbury talks about how latency can constrain the "thought" time – the maximum time in which a fully synchronised compute cycle (a "thought") can be completed. The amount that latency can be reduced is limited by the speed of light – the minimum thought time can be calculated by dividing the distance between the two furthest nodes by the speed of light, then multiplying by two. Bradbury's explanation of this can be found in [*Appendix C*].

---

[12] (Seabaugh, 2013) – An article on the challenges of shrinking MOSFETs (metal-oxide-silicon field-effect transistor). Although the article is old, the website is reputable, and the author is a professor of electrical engineering at the University of Notre Dame. Furthermore, there are links embedded throughout the article to support various statements.

[13] (Bradbury, 1999) – A paper that explores the topic of supermassive computers. The author is not particularly well known, but the paper is clearly reasoned and very well referenced. The paper only explores ideas, and doesn't offer any conclusive findings, therefore I believe it is reliable enough for me to carefully reference.

3.3 Alternative Computing Technologies

Although the popularity of MOSFET powered digital computers is undeniable, it remains to be seen whether they shall remain the pinnacle of performance. Alternative transistor types (such as TFETs) have been designed to bypass some of the inherent limitations of MOSFETs, which could potentially lead to smaller transistors (therefore higher density and greater performance). However, they are yet to gain popularity. Other techniques (such as optical computing) offer theoretically better performance but have been criticised for offering no significant real-world advantages.

One technique which shows promise is the domain of quantum computing. At a high level, quantum computers harness quantum mechanical phenomena to solve certain types of problems (notably integer searching and integer factorisation) exponentially faster than classical computers[14]. One specific area where classical Turing machines struggle is in quantum simulation – they experience exponential slowdown as the scale of the simulation increases. However, quantum computers do not experience such a slowdown and can therefore be used to simulate any local quantum system efficiently, as shown by Lloyd in his paper on universal quantum simulators[15].

3.4 Machine Learning

One of the quickest developing areas related to computer science is the field of machine learning. In particular, artificial neural networks have been improving at a rapid pace, growing in both size and complexity. ANNs are rudimentary simulations of the brain – they can learn from training data and then be used on unseen data. For example, a model called GPT-3 was trained to mimic human-written articles such that human readers could only identify them with 52% accuracy (little better than random guessing)[16].

Neural networks have a wide range of applications, from diagnosing brain scans to playing chess. One particular application that is being explored at the moment is the use of ANNs in intensive physics simulations. In their paper on modelling liquid splashes, Um et al. conclude that a combined approach of both traditional and machine learning techniques provided similar results to a high-resolution classical simulation with a fraction of the performance impact (6.5 times faster)[17]. Neural network simulations are empirical models, as they attempt to mimic (rather than exactly replicate) system behaviour.

[14] (Yanofsky, et al., 2008) – A textbook authored by two accomplished professors and published by a reputable university press. The textbook is thorough and well written, with an extensive bibliography.
[15] (Universal Quantum Simulators, 1996) – A paper that proves Feynman's 1982 conjecture that quantum computers can be programmed to simulate any local system. The author is a reputable professor at MIT who has written many papers and books on the subjects of complex (especially quantum) systems.
[16] (Brown, et al., 2020) – An extensive paper on natural language processing using the autoregressive language model GPT-3 with 175 billion parameters. The paper has 31 authors and is funded by reputable research company OpenAI. Although the paper is yet to be peer reviewed, it is very well referenced, and therefore I deem it reliable enough to use.
[17] (Liquid Splash Modeling with Neural Networks, 2017) – A paper by three researchers at the Technical University of Munich on the uses of CNNs to speed up the simulation of fluid-implicit particles.

4 Theoretical Physics

4.1 An Exact Model

In order for a model to predict exactly the future of a system, it must exactly replicate the behaviour of the entire system. There are two parts which make this challenging – the rules that govern the system must be exactly known, and every part of the system must be accounted for.

4.2 A Theory of Everything

A question that has plagued the minds of physicists for centuries is the question of whether we will ever be able to transcribe mathematically the workings of our universe in a "Theory of Everything" (ToE). Several attempts have been made, but so far, they have all given incomplete answers[18]. For example, classical mechanics (CM) is a simple theory that predicts the motion of a similar scale and speed to humans. However, the theory begins to break down as these bounds are exceeded. Notably, CM does not correctly predict the procession of the perihelion of orbits of Mercury: there is a 43 arcsecond per century error.

One current theory that models our universe is general relativity. General relativity (GR) is accurate in the domain of the large and the fast. Specifically, it generalizes special relativity (which deals with high speeds and moving frames of reference) and refines Newton's law of universal gravitation by describing gravity as the curvature of spacetime (a 4D unification of space and time). This allows it to be much more accurate than classical mechanics – unlike CM, it correctly predicts the motion of Mercury.

Another current theory that models our universe is quantum field theory. QFT combines classical field theory with special relativity and quantum mechanics, but not general relativity's description of gravity. It treats particles as excited states of underlying fundamental fields. QFT is most accurate at scales smaller from a fraction of a millimetre down to $10^{-19}\mathrm{m}$.

Whilst physicists have experimentally confirmed almost every prediction made by both these theories within their respective domains, they have yet to be unified into a theory of quantum gravity. The two main contenders at the moment are string theory and loop quantum gravity – advancements are being made in both, but (according to Carlip's review), there are still significant hurdles to overcome in each.

[18] (Quantum Gravity: A Progress Report, 2001) – A review of the progress that has been made in reconciling GR and QFT into a ToE. The Author is a professor at the University of California and is well known for his work on quantum gravity. The report is detailed, with over 350 references. The paper is quite old, however, and so advancements may have been made since then.

4.3 Fundamental Limits of Omniscience

It is impossible to store the exact state of the universe within the universe itself – this creates a recursive paradox, where the information store would have to contain itself. However, if the information store is sufficiently isolated from the rest of the universe, it may be possible to store the entire state of the rest of the universe (not including the storage itself).

4.3.1 Information Storage

In his 1997 paper on entropy in bounded systems[19], Bekenstein derived a formula for the upper limit on the information that can be contained within a given finite region of space which has a finite amount of energy – or, conversely, the maximal amount of information (I) required to perfectly describe a given physical system down to the quantum level. The equation is called the "Bekenstein Bound" [*Equation 1*], it and gives information in terms of radius (R) and mass (M).

Equation 1

$$I \leq \frac{2\pi cRM}{\hbar \ln 2} \approx 2.58 \times 10^{43} \cdot M \cdot R$$

The formula actually indicates that the highest density information store is the event horizon of a black hole, with each 4 Planck areas on the surface equal to one bit of information [*Figure 4*].

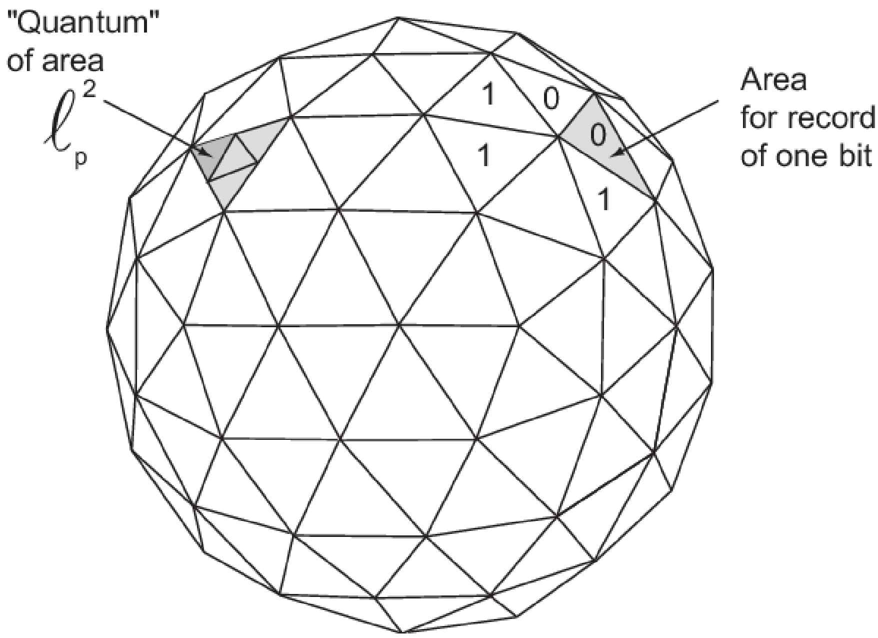


Figure 4 – A symbolic representation of the event horizon of a black hole (Bolotin, et al., 2020)

[19] (Universal Upper Bound on the Entropy-to-Energy Ratio for Bounded Systems, 1981) – A famous paper by Bekenstein on the link between energy and entropy in a bounded system. The author has a Doctor of Philosophy degree from Princeton University, and has received multiple awards for his work on black hole thermodynamics and other links between information and gravitation.

Using the values from NASA's website[20], we can calculate that the quantum state of all atoms (excluding dark matter and dark energy) in the universe can be expressed with an upper limit of:

$$I \leq 2.58 \times 10^{43} \frac{\mathrm{bits}}{\mathrm{kg} \cdot \mathrm{m}} \cdot \left(4.6\% \times 9.9 \times 10^{-27} \frac{\mathrm{kg}}{\mathrm{m}^3} \cdot 3.6 \times 10^{80} \mathrm{m}^3\right) \cdot 4.4 \times 10^{26} \mathrm{m}$$

$$\approx 1.9 \times 10^{123} \text{ bits}$$

We can substitute the equation for the Schwarzschild radius of a black hole [*Equation 2*] into the Bekenstein bound equation [*Equation 1*] and rearrange to find the mass of the black hole that must be created to store the equivalent amount of information [*Equation 3*].

Equation 2

$$R = \frac{2GM}{c^2}$$

Equation 3

$$I \leq \frac{2\pi c R M}{\hbar \ln 2}$$

$$I \leq \frac{2\pi c \left(\frac{2GM}{c^2}\right) M}{\hbar \ln 2} = \frac{4\pi G M^2}{\hbar c \ln 2}$$

$$M \geq \sqrt[2]{\frac{I \, \hbar c \ln 2}{4\pi G}}$$

By substituting the $I = 1.9 \times 10^{123}$ we obtained from the NASA values for the observable universe, we find $M \geq 2.2 \times 10^{53}$ kg – this is actually more than the energy of all the atoms in the entire universe ($\sim 1.8 \times 10^{53}$kg according to the NASA figures). The fact makes intuitive sense, as the amount of information stored in the position of the mass has been reduced (the radius of the space has decreased), therefore the amount of mass must increase to represent the same amount of information.

As a more practical comparison, the solar system's mass (1.992×10^{30}kg) can be encoded with 7.68×10^{86} bits. This equates to a black hole with mass 71200 times greater than that of the Sun (with a radius 33 times bigger than the Earth's). This is a size that, whilst still impractical, is technically possible to achieve.

[20] (NASA, 2014) – Information gathered by NASA's Wilkinson Microwave Anisotropy Probe about the composition of the universe. NASA is a world leading organisation, and so I deem the information on their website to be reliable enough to use in my research.

4.3.2 Measurement and Cloning

However, just because it may be possible to store that much information, doesn't mean that collecting the information is any easier. Heisenberg's uncertainty principle limits the amount of information we can know about the position and momentum of a particle: the more we know about one, the less we know about the other[21].

The "no-cloning" theorem elaborates on this[14] – it states that it is impossible to create an independent and identical copy of an arbitrary unknown quantum state. Although external sources on the idea could not be found, the theorem seems to imply that the bits of information which can be stored in the quantum state are only the maximum we can store / read from these states, and do not in fact fully describe the state. This perhaps provides an explanation of probabilistic phenomena – there might be underlying determinism, but we would never know enough about the underlying state, therefore the difference is indistinguishable.

[21] (Jones, 2018) – A webpage that covers an overview of the Heisenberg Uncertainty Principle. The page is not well referenced nor peer reviewed, and so I am being careful to use it as a general introduction to the topic (as opposed to a thorough resource). The author does have a degree in physics, however, so I deem the page reliable enough for limited use.

5 Discussion

At the beginning of this dissertation, the goal of answering the question "will predicting the future with computers ever be possible" was proposed. The three research sections provided background information surrounding the topic - this section shall set out to use that information by constructing a justified answer to the question.

5.1 Criteria for an Exact Universe Simulation

In order for a computer to ever be able to predict the future, I conclude that five conditions must be met.

A. There must exist a mathematical model that perfectly describes every aspect of our universe (a theory of everything).
B. The entire state of the universe must be perfectly described within the mathematical model's running memory.
C. The model must be able to run on the computer with perfect accuracy.
D. The existence of the computer must not affect the universe it is simulating, or it must take account for the effect of its existence.
E. The computer must be able to run the program so that simulated time is faster than the "real" time in which the computer operates.

5.2 Justification of Criteria

Criterion A must be met, otherwise there would be imperfections in certain parts of the simulation. These would appear small, but the chaotic nature of the universe would mean that small changes would compound over time, until the outcome would be entirely different. Criteria B, C, and D are justified with the same reason – failure to account for certain elements of the system or slight errors in calculation would cause the simulation to fall victim to chaos.

Criterion E must trivially be true – otherwise, the simulation would not predict the future, but the past (or present, if the simulated time exactly equals the real time).

5.3 Evaluation of Criteria

5.3.1 Criterion A

Although scientists have not found a unifying theory of everything, there are advancements being made in two major theories that are showing promise (string theory and loop quantum gravity). At the moment, I believe that we are likely to calculate or derive a ToE at some point in the future – therefore, criterion A will likely pass.

5.3.2 Criterion B

As described in section **4.3**, there are two parts to storing the complete state of the universe. The first part is to determine whether we can accurately measure the state of the universe, and the second part is to determine whether we can physically store the information that would be required.

Accurately measuring the state of the universe is challenging for three reasons. Firstly, the universe is incredibly large. This means that the process of measuring is infeasible. Furthermore, the measurements must all be taken simultaneously, or chaos will once again take hold. One positive point to note is that, as far as we know, causality cannot be violated due to the limit of the speed of light. This means that we should only need to know information about the state of the universe within the sphere of influence for however long we want the simulation to run (e.g. predicting 1 minute into the future would require knowing the state of the universe up to 1 light-minute away). However, it is unclear how quantum entanglement fits into this model: if every part of the universe is in some way entangled, it may be necessary to know the entire state.

In addition, it is unclear whether the bits of information we can measure from a quantum state completely describe it, or whether there is some hidden state that causes the evolution of the state to appear probabilistic. Furthermore, it is unclear whether the "no-cloning" theorem prevents the exact quantum state from being copied.

Storing the state of the universe is also a challenge, due to the massive amounts of information measured. As discussed in section **4.3.1**, Bekenstein discovered a formula for the maximum amount of information that can be contained within a given volume. He also discovered that the maximum information density can only be achieved with a black hole – the information that can be stored is proportional to the surface area of the black hole. This means that, theoretically, a black hole could be created such that the information on the surface area described the state of the rest of the universe. However, extracting the information may prove difficult, as scientists are still trying to figure out how the information is encoded (the most popular theory is that it is released via Hawking radiation, but no specific algorithm for decoding the information has been proposed).

Because of the significant challenges posed in this criterion, I do not think that storing an exact copy of the universe is feasible, even if it may be possible.

5.3.3 Criterion C

Whether the theoretical computer would be able to run the simulation accurately depends on the properties of whatever ToE is discovered (if one is discovered of course).

If the model implies that the universe has a finite number of states which are all theoretically measurable, then it can be considered discrete, and therefore ideal for a Turing machine to simulate.

If the model implies that the universe has an infinite number of knowable states, then it can be considered continuous. A continuous model is impossible for a Turing machine to exactly simulate, as the infinite majority of states would be impossible to represent in finite memory (see section **3.1**). However, it can be relatively easy to approximate.

If the model's underlying state can never be completely known, then it is considered uncertain. This is hard for a classical computer to simulate, as the number of possible outcomes to simulate grows exponentially. Quantum computers outperform classical computers in this area, as they can efficiently solve certain types of problems with exponential complexity.

To complicate matters, exactly simulating continuous time is impossible regardless of the universe's measurability (unless the differential equations describing the model can be solved). Thankfully, general relativity implies the intrinsic linking of space and time into a unified spacetime – it is unlikely that one is finite and the other continuous.

At the moment, the most popular theory is that our universe is both continuous and uncertain (QFT). If this is the case, then it would be improbable for a computer to run the simulation accurately. However, one theory that's gaining popularity is the holographic principle – a property of string theory that is exhibited by the Bekenstein bound. The holographic principle suggests that our three-dimensional universe is actually a two-dimensional surface of information. If true, this theory could potentially allow for easier computation than QFT.

Overall, I think that it is possible (but not certain) that criterion C will pass.

5.3.4 Criterion D

In order for the existence of the computer to not affect the accuracy of the simulation, it must be completely isolated. Alternatively, it must be able to simulate the interactions of itself with the outside universe. Without directly simulating itself (which would cause an infinite recursive computation during a timestep), the only way to do this is to be able to exactly determine / control the interactions between the computer and the surrounding universe.

The only solution to either of these that I can think of would be to place the computer in some hypothetical container where interaction between the inside and outside is simplified or eliminated. An interesting idea to note is that the very action of outputting the prediction from such a container could cause reality to diverge from the simulated path (due to chaos).

I do not think that this criterion will ever be satisfied, due to the immense challenge of isolating the computer from the outside universe.

5.3.5 Criterion E

Creating a computer powerful enough to run a universe simulation at faster than real time would be massive challenge.

Firstly, in the classical model, doubling the number of objects in the universe quadruples the number of interactions – this causes the intensity of the simulation to grow rapidly as the size of the simulation increases. Although the introduction of quantum effects causes the calculations to increase exponentially, the use of quantum computers can then negate the effect. The true computational cost of a universe simulation is unknown, but one overwhelming likelihood is that it will require a Matrioshka brain to simulate. Currently, I think that more knowledge about the properties of the ToE is required before the feasibility of constructing such a device is evaluated.

5.3.6 Summary

Overall, I think that an exact simulation of the universe is a long way off. Although I believe that discovering a ToE and constructing a suitable computer (with accompanying storage facility) is likely, I do not believe that we will ever be able to measure accurately the entire state of the entire universe. Furthermore, I think the challenge of isolating the computer from the surrounding computer is too great without a breakthrough in physics to make it easier.

5.4 Accepting Inaccuracy

In section **5.3.6**, I concluded that it was very unlikely that an exact universe simulation would ever be possible. However, if we accept that inaccuracy will always creep in, we can use a variety of techniques which can provide relatively accurate results for a fraction of the computational resources.

One such optimisation is to simulate accurately only a bounded area, and then approximate the interactions between the bounded area and the outside world. For example, accurately simulating an electronic device, but approximating the power supply or the cooling system. This means that focus areas can be accurately simulated, whilst less important parts can be approximated.

This can be taken a step further by approximating even the focus areas. It is widely accepted that the universe is an example of a complex system: one feature of complex systems is that there is disorder (chaos) at the micro level, but robust order at the macro level. This means that low level quantum interactions can be abstracted away, and a suitable simplified mechanistic model can be implemented. Simplified models can be much easier to compute, and do not require the exact state to be measured, therefore they are more practical in the real world.

For situations where mechanistic approaches are impractical, empirical models can be used. One particular situation is when simulating the interactions of people - artificial neural networks can be used to roughly approximate human behaviour. Training such networks is not particularly challenging in modern society, as mass amounts of data are already being collected and analysed by tech companies (so that they can provide personalised advertising).

As discussed in section **2.6**, ensemble forecasting can be used to reduce the impact of chaos. By combining results from several runs of the simulation (each with strategically varied initial conditions), the impact of anomalous runs can be reduced. This is particularly useful in stochastic models (such as when predicting the outcome of an evening of gambling), or in models where measurements are highly inaccurate (such as in weather forecasting).

6 Conclusion

6.1 Findings

In the introduction, I set out to answer whether a perfect universe simulation was possible. I now believe that such a simulation is mostly likely too impractical to attempt, at least without a major breakthrough in physics to allow for the computer to be isolated from the surrounding universe. However, various techniques can be used to approximate the universe by ignoring disorder at the micro level (and instead modelling emergent phenomena at the macro level) – such techniques can provide sufficient accuracy for most practical uses, but chaos will cause them to become more and more inaccurate as time progresses.

6.2 Impact of Ideas Discussed

Some of the ideas introduced in this dissertation have ethical and philosophical impacts that have not been discussed so far.

Firstly, the ability to simulate perfectly the universe means that the universe must be deterministic – an idea that has profound impacts on philosophy. It would provide definitive evidence that free will is an illusion, as well as some evidence in support of the emergent theory of consciousness.

Secondly, there are ethical issues with simulating consciousnesses. One such example is the question of whether it is ethical to bring beings into existence which are destined to suffer: should we value real suffering more than simulated suffering? Furthermore, is it possible that we are living in a universe simulation?

There are also issues regarding the use of such a technology. One obvious use is to prevent crimes from ever occurring, by arresting offenders before they commit the crime. A question this raises is whether people should be judged by actions that they have not yet committed – can it be justified in the name of public safety?

6.3 Limitations of Methodology and Opportunities for Further Research

During the process of writing the dissertation, I attempted to choose the most reliable sources I could find. However, often I had to decide between a reliable source which was overly complex, and a less reliable source that was easier to comprehend. See *Figure 5* for a chart which shows the types of sources that I used.

I deem the peer reviewed articles to be of very high reliability, the unreviewed articles and books to be quite reliable, and the websites to be of varying reliability. Overall, I think that I made good choices when balancing source complexity with reliability – I was careful in my usage of potentially unreliable sources, and so I don't think that any reliability issues are likely to affect my conclusion.

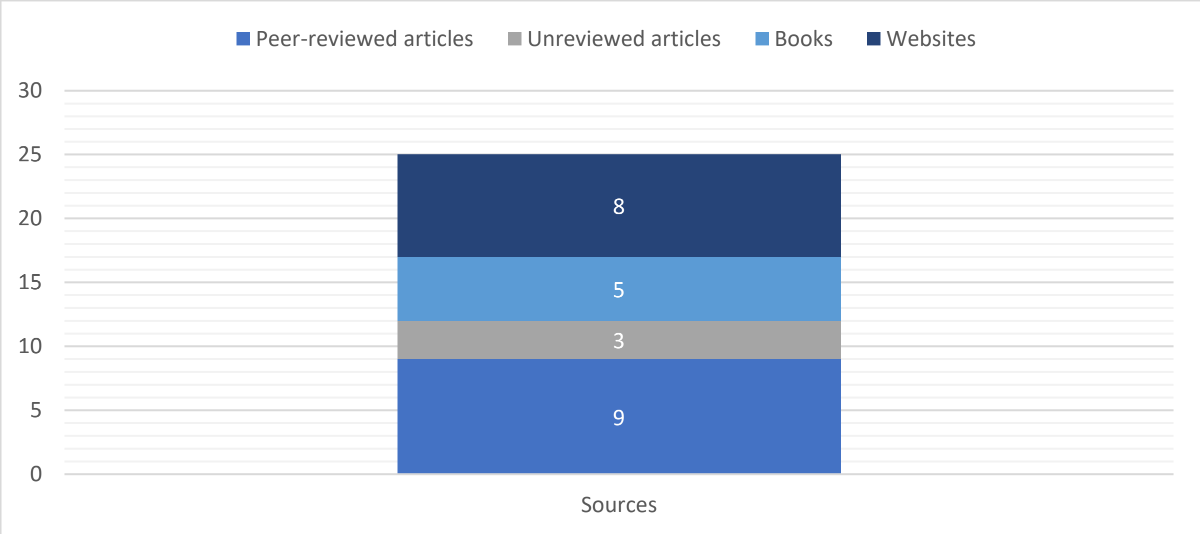


Figure 5 – Breakdown of sources

In addition to source reliability, content covered also poses a problem to the project methodology. Many topics have been overviewed but not deeply researched:

1. I touched on our inability to solve the equations of the three body problem – I didn't elaborate on the numerical techniques which can approximate them (let alone proofs for the inherent inaccuracy of such techniques).
2. I believe that further research is needed on the effectiveness of ensemble forecasting using chaotic models. I have assumed in section **5.4** that the technique is equally effective on chaotic and non-chaotic systems, but this may not be the case.
3. In section **3.2**, I mentioned Matrioshka brains, but did not expand in any specifics. I think more research could be done to explore the possibilities of their construction and operation, in order to evaluate the feasibility of their construction.
4. In section **3.3**, I touched on the concept of quantum computing, but did not research further into the actual engineering of such devices. This means that the engineering challenges in constructing a quantum Matrioshka brain may not be the same as those of constructing a classical one.
5. More research could be done on the potential for neural network based crowd simulations to predict general trends in society (the outcome of elections, for example). This could allow forecasting to be much more computationally efficient than traditional mechanistic modelling.
6. The ability to perfectly simulate parts of the universe poses many philosophical and ethical questions – I have mentioned a few in section **6.2** but believe that more discussion is needed on the topic.

Because of these limitations, there may be some parts of my reasoning that are inaccurate, which could perhaps change the conclusion of the dissertation. However, I deem most of the research reliable enough to be confident in the findings presented in section **6.1**.

7 Evaluation

7.1 Project Aims

Overall, I think that my project succeeded at fulfilling the goals I attempted to achieve. At the start of the project, I aimed to answer four questions:

1. How can we mathematically describe our universe?
2. Is it possible to exactly simulate the future using a computer?
3. What would such a computer look like?
4. Would it be feasible to create?
5. What techniques can improve performance, and are there any trade-offs?

I believe that that aims 1, 2, and 4 were definitively answered in sections **5.1**, whilst aim 3 was described in section **3.2**. For brevity's sake I chose not to fully answer aim 5, as I deemed the topic too large to be able to fit within this dissertation. I did, however, touch on the question on some of the broader ideas related to optimisation in section **5.4**.

One of the reasons I chose the topic of computational forecasting was that it allowed me to gain knowledge and understanding of topics that I find particularly interesting – it ties together cutting-edge theoretical physics, high performance computing, and interesting mathematical concepts. I believe that I have succeeded in this regard – I am much more confident in my ability to answer questions about these areas. The knowledge I have gained is not only of personal interest – I deliberately chose areas to research (such as computability and quantum computing) that should help me when studying for a degree in Computer Science.

7.2 Process

Overall, I think my process of completing the project was good: I produced a dissertation that I'm proud of, increased my knowledge of the subject area, and did it all whilst being largely self-guided. However, there are still some lessons that I have learnt from:

7.2.1 "Productivity" Tools

One of the things I planned during the initial stages of my project was the productivity software that I would use. However, I found myself often spending more time organising these tools than actually working on the project. Using such software gave me the feeling of being productive, whilst actually getting very little words on the page.

Next time I attempt a large project, I shall make sure to be very selective when choosing tools to use, and I will periodically do cost-benefit analyses to ensure that I am not wasting time.

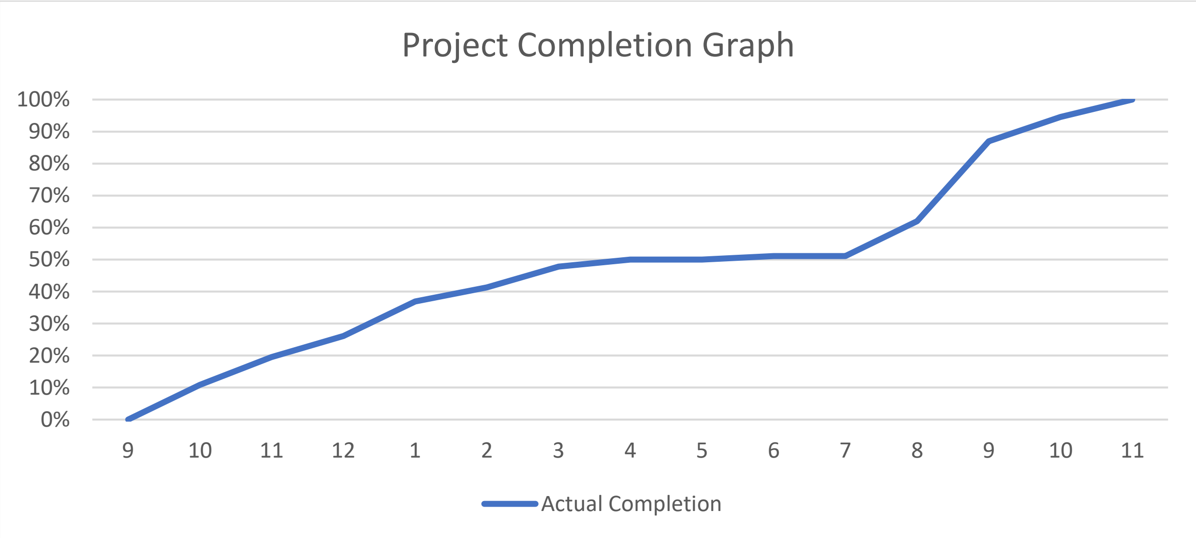
7.2.2 Dunning-Kruger Effect

According to Wikipedia, the Dunning-Kruger effect is "a cognitive bias in which people with low ability at a task overestimate their ability". I encountered this significantly during the planning and general research stages, where I often overestimated my knowledge in particular areas – in other words, I didn't know what I didn't know. This means that I often underestimated how much I would need to write to cover some topics, which overall led me to initially propose a question much broader than would have been practical given the constraints of the project. Once I realised this, however, I swiftly adjusted the question and narrowed my focus.

If I were to do a similar project again, I will make sure to do more research during the planning stage, to help offset my initial biases.

7.2.3 Time Management

Time management is one of the areas that I have learnt most from this project. Below is a graph of my estimated project completion for each month. I created it by using my project logs to estimate a "productivity score" out of 10 for each month, and then plotting a cumulative graph in terms of the productivity score total.



The progress on the project was consistent from September to March, at which point I had finished almost all of the research I needed to do to begin writing the dissertation. However, it was during March that the UK went into lockdown, and the combination of uncertainty combined with lack of pressure caused my progress to plateau. Around July, I set myself a personal deadline – finish the dissertation by the time school starts or drop the project to prevent it from interfering with my school grades. This caused my productivity to sharply increase, resulting in the bulk of the dissertation being finished by the time school started in September.

This has taught me that personal deadlines work surprisingly well for me – I shall make sure to use them more frequently when tackling large projects.

Appendices

Appendix A

Firstly, sensitivity to initial conditions is the property discovered by Lorenz – the butterfly effect. It says that slightly changing the state will lead to drastically different results as the system progresses. Slight changes will appear to initially follow the same path of the original – however, they will quickly devolve into drastically different outcomes.

Secondly, topologically transitive means that the states of a system quickly become "mixed up". For example, $f(x) = 50^x$ is very sensitive to initial conditions (x), but it is not topologically transitive as the outputs do not get mixed up. Formally, a continuous map $f : S \to S$ is said to be topologically transitive if, for every pair of non-empty open sets $A, B \subset S$ there exists an integer n such that

$$f^n(A) \cap B \neq \emptyset$$

where f^n is the nth iterate of f, and $\emptyset$ is the empty set.

The final property is dense periodic points. A mapping $f(q,t)$ describing the state of a system with initial state q at time t is defined as having dense periodic orbits if, for any state, there is an arbitrarily close point p such that:

$$\exists \Delta t : f(p, 0) = f(p, \Delta t)$$

In other words, a system at state p will eventually end up back at p.

Appendix B

A deterministic Turing machine M can be formally defined as a 7-tuple: $M = \langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$ where:

- Q is a finite, non-empty set of states;
- Γ is a finite, non-empty set of tape alphabet symbols;
- $b \in \Gamma$ is the blank symbol (the only symbol allowed to occur infinitely often on the tape);
- $\Sigma \subseteq \Gamma - \{b\}$ is the set of input symbols (the set of symbols allowed to appear on the initial tape contents);
- $q_0 \in Q$ is the initial state;
- $F \subseteq Q$ is the set of final states;
- $\delta: (Q - F) \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is the transition function that takes the current state and the scanned symbol and returns (according to a state table) the next state, the symbol to write, and the direction to move the head (left or right).

Appendix C

Robert Bradbury on the latency and thought time of Matrioshka brains (MBs):

> *"This is clearly seen when imagining the management of three different planetary probes, one on the moon, one on Mars and one orbiting Saturn. The moon probe may be managed from earth in real time. The Mars probe can be given directions between cups of coffee. The Saturn probe can be given directions only several times a day. If you expect the more distant probes to do useful work in a reasonable time you have to build into them increased amounts of intelligence and autonomy.*
>
> *If the thoughts between CPUs in a MB are independent, then the brain can be made very large with little effect. If, however, the MB is attempting to solve a problem which requires all of its capacity then it must think slower to maintain synchronization between CPUs as their inter-node distance increases. In theory MBs orbiting in the galactic halo, a KT-III civilization would be able to think collectively, but their "thought" time must be on the order of tens of thousands of years or more."*

Glossary

artificial neural network – a machine learning technique inspired by the structure of biological brains.

chaos – the exponentially compounding effects of slight variations in initial conditions to entirely different final states.

classical mechanics – a physical theory that is accurate enough at describing the motion of macroscopic objects. However, inaccuracies emerge at the extremities of velocity, distance, and mass.

clock speed – the frequency of processing cycles within a CPU.

computable numbers – numbers that can be computed to within any desired accuracy by a Turing machine given a finite, terminating tape.

CPU – central processing unit of a computer, responsible for logical and arithmetic operations.

differential equation – an equation that describes the rate of change of a variable. Sometimes, sets of differential equations can be solved to give further insight into their behaviour. Other times, they must instead be approximated through numerical methods.

emergence – emergence is when structure appears to emerge from disorder. It is often found in complex systems.

ensemble forecasting – a method that involves running a simulation multiple times with strategically varied initial conditions in order to gain insight into the distribution of possible outcomes.

general relativity (GR) – one of the two most popular scientific theories. Proposed by Einstein, the theory deals with massive objects and large distances.

graph – a set of vertices that are connected by edges. Graphs can be used to show the interactions between items in a system (where vertices are items and edges are interactions).

Heisenberg's uncertainty principle – a theory that our ability to measure the state of a system is fundamentally impaired: the more we know about a particles momentum, the less we know about its position (and vice versa).

information – a complex concept which attempts to measure the amount of "surprise", "uncertainty", or "entropy" in a variable's possible outcomes. The unit is usually bits – the average number of yes or no questions that would be needed to describe fully the state of the system.

loop quantum gravity – an attempt to merge QFT and GR, competes with string theory as a candidate for a quantum gravity.

machine learning – a collection of techniques that allow computers to learn patterns in data, which can then be used to make predictions about patterns in unseen data. The most common method of machine learning uses rough approximations of brains called "neural networks" to achieve this.

mathematical model – a way of summarising the behaviour of variables (or an entire system) using the language of mathematics.

> **empirical model** – a model that approximates large patterns that emerge from the system.
>
> **mechanistic model** – a model that describes detailed interactions between elements in a system.
>
> **stochastic model** – a model that describes behaviour of a system using probability distributions.

Matrioshka brain (MB) – theoretical supercomputer on the scale of stars, galaxies, and universes.

memory – the maximum amount of information that a computer can store as it is running.

no-cloning theorem – the theorem states that it is impossible to create an independent and identical copy of an unknown quantum state.

quantum computer – a computer that uses quantum phenomena to increase exponentially the speed of solving certain problems.

quantum field theory – one of the two most popular scientific theories. The theory deals with the behaviour of tiny particles and very small distances.

simulation – a program that mimics the behaviour of something else (in this dissertation, a simulation of reality).

string theory – a theoretical framework in which point-like particles are replaced by one-dimensional, vibrating objects called strings.

system – a regularly interacting or interdependent set of items forming a unified whole.

> **complex system** – an ensemble of many elements which are interacting in a disordered way, resulting in robust organisation and memory.

three-body problem – the problem of taking the initial positions and velocities of three point masses and solving for their subsequent motion (according to Newton’s laws of motion and universal gravitation).

theory of everything (ToE) – a hypothetical mathematical model that describes every behaviour of the universe exactly.

Turing machine – a formal way of representing a digital computer, used to mathematically prove theories related to the idea of computation and complexity.

Bibliography

Benbennick, David. The complete graph on 7 vertices. *Wikimedia.* [Online] https://commons.wikimedia.org/w/index.php?curid=511711.

Bolotin, Yu. and Yanovsky, V. 2020. Physics of Limit Values at Planck Scale. *ResearchGate.* [Online] 2020.

Bradbury, Robert. 1999. Matrioshka Brains. *gwern.net.* [Online] 1999. https://www.gwern.net/docs/ai/1999-bradbury-matrioshkabrains.pdf.

Brown, Tom, et al. 2020. Language Models are Few-Shot Learners. *arXiv.org.* [Online] 2020. https://arxiv.org/pdf/2005.14165.pdf.

CSL4U. Computer Science Principles - Turing Machine. *Open Book Project.* [Online] http://www.openbookproject.net/books/StudentCSP/CSPTuring/turingMachines.html.

Deterministic Nonperiodic Flow. **Lorenz, Edward. 1963.** s.l. : Journal of the Atmospheric Sciences, 1963, Vol. 20.

Devaney, Robert. 2003. *Intro to Chaotic Dynamical Systems.* s.l. : Westview Press, 2003. p. 180.

Dschwen. A solution in the Lorenz attractor rendered as an SVG. *Wikimedia.* [Online] https://commons.wikimedia.org/w/index.php?curid=494694.

Jones, Andrew. 2018. Understanding the Heisenberg Uncertainty Principle. *ThoughtCo.* [Online] 2018. https://www.thoughtco.com/the-heisenberg-uncertainty-principle-2699357.

Liquid Splash Modeling with Neural Networks. **Um, Kiwon, Hu, Xiangyu and Thuerey, Nils. 2017.** s.l. : Computer Graphics Forum, 2017, Vol. 37.

Marion, Glenn and Lawson, Daniel. 2008. *An Introduction to Mathematical Modelling.* s.l. : University of Bristol Website, 2008.

Merriam-Webster. *Merriam-Webster Online Dictionary.* [Online] https://www.merriam-webster.com/.

NASA. 2014. What is the Universe Made Of? [Online] 2014. https://map.gsfc.nasa.gov/universe/uni_matter.html.

Newton, Isaac. 1687. *Philosophiae Naturalis Principia Mathematica.* [trans.] Andrew Motte (1729). 1687. pp. 71-72.

On Computable Numbers, with an Application to the Entscheidungsproblem. **Turing, Alan. 1937.** s.l. : Proceedings of the London Mathematical Society, 1937, Vols. s2-42.

Quantum Gravity: A Progress Report. **Carlip, Steven. 2001.** 8, s.l. : Reports on Progress in Physics, 2001, Vol. 64.

Seabaugh, Alan. 2013. The Tunnelling Transistor. *IEEE Spectrum.* [Online] 2013. https://spectrum.ieee.org/semiconductors/devices/the-tunneling-transistor.

Stochastic Dynamic Prediction. **Epstein, Edward. 1969.** 6, s.l. : Tellus, 1969, Vol. 21.

Tetlock, Philip and Gardner, Dan. 2016. *Super-forecasting.* s.l. : Random House Books, 2016.

The Computation of π to 29,360,000 Decimal Digits Using Borweins' Quartically Convergent Algorithm. **Bailey, David. 1988.** s.l. : Mathematics of Computation, 1988, Vol. 50.

Universal Quantum Simulators. **Lloyd, Seth. 1996.** 5278, s.l. : Science, 1996, Vol. 273.

Universal Upper Bound on the Entropy-to-Energy Ratio for Bounded Systems. **Bekenstein, Jacob. 1981.** 2, s.l. : Physical Review D, 1981, Vol. 23.

What is a Complex System? **Ladyman, James, Lambert, James and Wiesner, Karoline. 2013.** s.l. : European Journal for Philosophy of Science, 2013, Vol. 3.

Wolfram, Stephen. 2002. *A New Kind Of Science.* s.l. : Wolfram Media, 2002. p. 972.

Yanofsky, Noson and Mannucci, Mirco. 2008. *Quantum Computing for Computer Scientists.* s.l. : Cambridge University Press, 2008.