School of Science and Technology

Mathematical Modelling In Finance

Alice Rudd
Supervisor: Dr. David Chappell
Department of Physics & Mathematics, Nottingham Trent University

Introduction:

In this project we are looking at how the Black Scholes Model can be used to predict European options.
An option is a contract which depending on its type, gives the holder the opportunity to buy/sell an asset such as a stock for a set price decided on the initial date to be exercised at a maturity date .There are two types of options:

- a put option gives the seller the right but not the obligation to sell the product
- a call option gives the right but not option the right to buy the option.

There are various other ways of making options, two common ones are American options and European options:

- An American option can be exercised at any time throughout the duration of the contract up until the maturity date
- European options can only be exercised on the Maturity date.

In order for the Black Scholes PDE to be used on these options, they need to satisfy a set of assumptions, these are; 1. that no dividends are paid during the life of the option, 2. that the movements of the markets can not be predicted, 3. when buying the option there are no transaction costs, 4. The option is only European and therefore can only be exercised on the date of expiration, 5. The assets returns are normally distributed with log and 6. the volatility and the risk-free interest rate are constant and are known.

Black Scholes Equation :

The Black Scholes Model (BSM) estimates the price of financial instruments, usually options. The BSM is only used for European Call options:

The PDE of the Black Scholes equation is [1]:

$$\frac{dC}{d\tau} - \frac{1}{2}\sigma^2 S^2 \frac{d^2C}{dS^2} - rS\frac{dC}{dS} + rC = 0,$$

and the exact solution to the Black Scholes PDE are given by the equations:

$$C(S,t) = SN(d_1) - Ee^{-p(T-\tau)}N(d_2)$$

$$\text{with:} \quad d_1 = \frac{\log\left(\frac{S}{E}\right) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \quad \text{and} \quad d_2 = d_1 - \sigma\sqrt{T-t}$$

Where:

- C= price of the Call option
- S= current stock price
- r= risk-free interest rate
- σ= volatility over time (standard deviation)
- E= Strike price
- $\tau = T-t$, with T being the expiration time and t being the current time.

European Options :

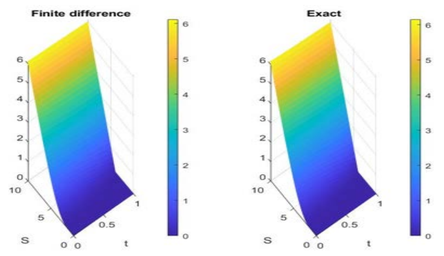
In order to get the code to model European Options in MATLAB, we took the Black Scholes PDE and applied the finite difference formula to each derivative in the equation.

To form the complete code this new equation was used along with the Boundary conditions that are required for a European Call option which are:

$$C(0,\tau) = 0 \quad \text{and} \quad C(S^*,\tau) = S^* - Ee^{-r\tau},$$

where $S^*$is a very large value of S as $S \to 0$.

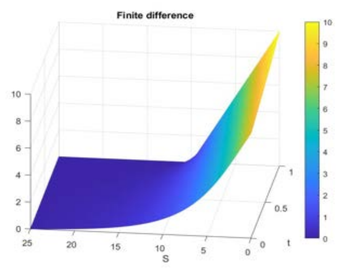
We compared this result to the Black Scholes formula above [1] to verify the code.



To model European Put options (P) instead, we use the finite difference code but the boundary conditions are changed to those needed for Put options:

$$P(0,\tau) = Ee^{-r\tau} \text{ and } P(S^*,\tau) = 0\,.$$

The result is similar to the Call option graph but goes in the opposite direction. The maximum is at 0 instead of the end point and the Value of the option tends to zero as it moves to the end point.



We used the finite difference method again since it is easily generalised for Put options (unlike the exact solution), and even extended to more complicated cases such as American options, as follows.

American options:

For American options we need to find a way to change what we apply the Finite difference method, so that the code accounts for the fact that the option can now be exercised at any date up until and including the Maturity date.

In order to make it so we had a code that worked for American Put options we used the Successive overrelaxation method (SOR), which is an iterative method of solving linear systems and is an extension of the Gauss-seidel Method [2].

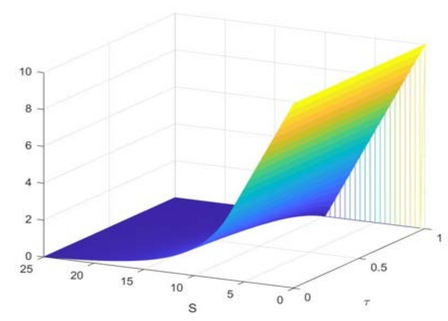
The boundary conditions for an American Put option are given by [2]:

$$P(0,\tau) = E \quad \text{and} \quad P(S^*,\tau) = 0.$$

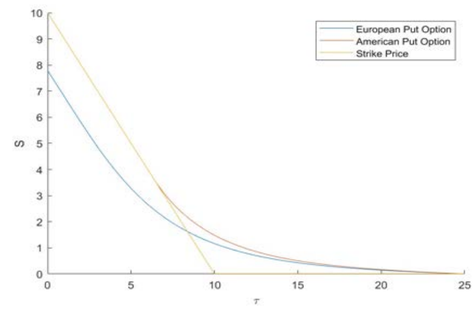
A key step in the code needed to create these graphs is the addition of a max function, which makes sure that the option never drops below *0*

$$v = \max(E - S, 0).$$

This graph shows the prediction of the American put options given by the code created using the SOR method:



The graph bellow shows the comparison of American Put options and European Put options against the strike price.



We can see that the price of American Put options never go bellow the Strike price of the option but the European put option curve does.

Conclusions:

- We can see that there are a variety of methods which can be used to predict and model different types of options
- Using the BSM we can see that in simple cases there are ways that we can find an exact solution, in this case for European options.
- From this work we can see that there are methods like the finite difference method that can easily be adapted to work for other types of options.

References:

[1] D. J. Higham, Black-Scholes for Scientific Computing Students, *Computing in Science and Engineering*, **6**, 2004, 1-10
[2] R. U. Seydel, *Tools for Computational Finance* (Springer, Berlin, 2009)

NOTTINGHAM TRENT UNIVERSITY

Mastering the Curriculum: Investigating the Impact of Learning for Mastery Assessments in Undergraduate Learning

Anya Scarlet Rose Supervisor: Dr. Matt Tranter

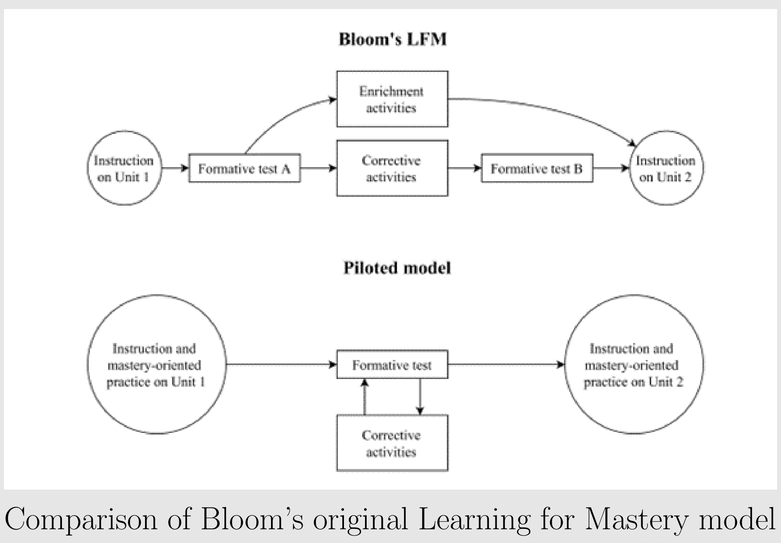
Department of Physics & Mathematics
Nottingham Trent University

Introduction

- The concept of "Learning for Mastery" (LFM) was initially popularised by Bloom in a 1968 paper [1].
- There are a range of abilities in classes. Giving all students the same amount of time and the same resources resulted in most students not mastering the study material.
- The steep ratio between teachers and students presented a challenge in practically implementing LFM. This can be overcome by taking advantage of technological advances.
- However, the usefulness and practicality of many of these methods are not often researched.
- LFM tests are used in the first-year mathematics module Mathematical Methods and this project investigated the use of these tests in the 2021/2022 cohort, analysing the effect, if any, of LFM tests on the student's exam performance.

Modern Mastery with CAS

In order to address the need for a teaching method which allows each student to have as little or as much practice as needed, we turn to a teaching cycle developed for the application of Computer Algebra Systems (CAS) in LFM. [2]



Comparison of Bloom's original Learning for Mastery model with the modern CAS adjusted version. [2]

With this modernised model, it is much easier to see how CAS could be used to take advantage of the repetitive need for more formative assessments. CAS can alter and process mathematical expressions directly without needing every expression's numerical value. We use CAS to randomly alter framework questions to create new formative assessments testing the same skills, with no further input needed from the assessor. The system used to create our LFM tests is Numbas. Our LFM tests have the following features:

- No time limit.
- Each answer can be checked in real time
- Each answer can be changed and re-checked an unlimited amount of times.
- The final mark achieved is not negatively affected by number of attempts at answers or time taken.

These features help to put emphasis on the mastery and perfection of each method, rather than the ability to answer fast and move on even when the content isn't fully understood.

Discussion & Future Work

With all the analysis performed, there is overwhelming evidence that the LFM tests had an important role in how the students performed. The LFM tests were significant, not just in the overall exam performance, but also in each individual topic. Being the most significant factor, even compared to in-person attendance, online engagement or grade profiles, shows the importance of their use. The system used to create them is easy to navigate, and if found to be significant in other modules, there is considerable motivation to use them in more applications. Although CAS systems lend themselves to STEM subject questions, the premise of the LFM test is applicable in every department, with multiple-choice or worded answers being possible in contrast to the typical numerical or algebraic answer.

There have been instances of other departments within STEM using a CAS system for mastery, specifically in an engineering undergraduate degree [2], which also resulted in positive results. It would be greatly beneficial to trial these tests in other modules or departments to analyse their potential on a larger scale. Even without the aim of further analysis, these tests help students get closer to mastery, and that should be a great motivation to implement them further.

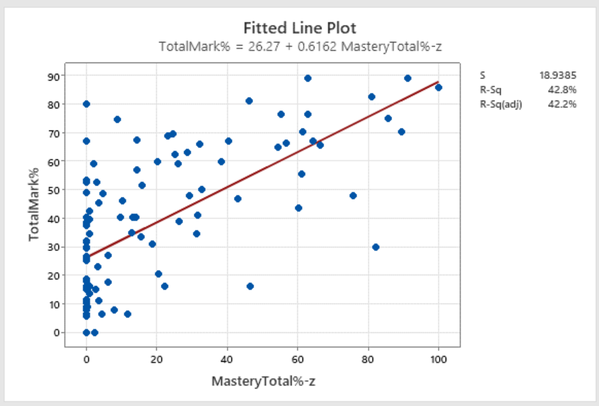
Hypothesis & Testing

To determine the effect, if any, that the LFM tests had on the student's exam performance, we decided on two main lines of statistical testing:

- If the students who used LFM tests had significantly better exam results than those who did not, tested using a 2 sample T-test.
- If LFM results achieved by a student were a significant positive contributing factor to said student's exam results, investigated using regression.

With these two tests in mind, we first analysed the overall connection between the use of any LFM tests and the overall exam performance of students.

Here we see the scatter plot of mastery test scores and exam scores, with the related line of best fit using regression. Both the 2 sample t-test and the regression analysis resulted in the most significant results possible, with a p-value of 0 in each case.



After considering the overall LFM test usage and exam performance, we also saw fit to test all of the modules' topics separately. As the LFM tests were created to correspond with the module, each topic had a corresponding LFM test and exam question(s). Although this lowers the sample size due to fewer students engaging in every LFM test separately, the samples all remained considerable, with at least 30 students' data being used.

	MELS	FLD	PDO	I	CN	ODE	MOD
2-sample t-test p-value	0	0	0	0	0.013	0	0.001
Regression analysis p-value	0.007	0.116	0.001	0.049	0.001	0.006	0.045

As shown in the table above, all but one test resulted in a significant result. This strongly supports the module wide observations.

Modelling Exam Performance

Rigorously investigating the connection between the LFM test and increased exam performance is difficult. There are arguments to be made that students who engage in the LFM tests already show they have a level of motivation higher than the majority of their peers. This behaviour alone could be seen as a cause for higher exam performance rather than it being the LFM tests themselves. To be sure of the LFM tests' significance, especially compared to other contributing factors, we used Multi-Level Regression to model exam performance with all data available to us. Considering online video engagement is crucial for this module specifically as all the topics had videos pre-recorded online, with the primary instruction being to watch the videos before the lectures, as a main introduction to the material, with the lectures being to answer questions and clarify. Below is the predictors used within the model with their associated p-value.

Predictor	p-value
Total minuted delivered of online videos	0.466
Total views and downloads of online videos	0.695
UCAS tariff points	0.975
Taken Maths A-level (True/False)	0.009
Taken Further Maths A-level (True/False)	0.600
Mathematical Methods attendance	0.041
Course-wide attendance	0.002
Widening participation student (True/False)	0.617
LFM test total percentage	0.000

From these results, we can see the most significant predicting factor is the Mastery test results, closely followed by the students overall attendance, if the student has previously taken a Maths A-level, and lastly the module attendance itself.

References

[1] Benjamin S. Bloom, *Learning for Mastery*, Evaluation Comment **1**(2), 1 - 12 (1968).

[2] Timo Pelkola, Antti Rasila, Christopher Sangwin, *Blended Mastery Learning in Mathematics*, Informatics in Education **17**(2) 363 - 380 (2018).

Initial-Value Problem for a System of Boussinesq-Type Equations

Ben Mardell N0800431

Department of Physics & Mathematics
Nottingham Trent University

INTRODUCTION

Boussinesq-type equations are an approximation to the **Navier-Stokes equations** which govern the fluid dynamics of waves on a small scale. They are particularly useful in coastal engineering and computer modelling of shallow waves [1]. They can also model longitudinal waves in layered waveguides [2].

In this study we will look at the **system of three Boussinesq-type equations** with **regular coupling**. The initial-value problem for this system, in **epsilon form**, is

$$\begin{aligned} u_{tt} - u_{xx} &= \varepsilon\left(\frac{1}{2}\left(u^2\right)_{xx} + u_{ttxx} - \delta_1\left(u - w\right)\right), \\ w_{tt} - c_1^2 w_{xx} &= \varepsilon\left(\frac{\alpha_1}{2}\left(w^2\right)_{xx} + \beta_1 w_{ttxx} + \gamma_1\left(u - w\right) - \delta_2\left(w - z\right)\right), \\ z_{tt} - c_2^2 z_{xx} &= \varepsilon\left(\frac{\alpha_2}{2}\left(z^2\right)_{xx} + \beta_2 z_{ttxx} + \gamma_2\left(w - z\right)\right). \end{aligned}$$

However, we will also introduce a **scaling** to the system as follows; $x = \sqrt{\varepsilon}\tilde{x}$, $= \sqrt{\varepsilon}\tilde{t}$, $u = \frac{1}{\varepsilon}\tilde{u}$, $w = \frac{1}{\varepsilon}\tilde{w}$ and $z = \frac{1}{\varepsilon}\tilde{z}$. It is clear that introducing this scaling leads to the following system of equations (omitting the tildes)

$$\begin{aligned} u_{tt} - u_{xx} &= \frac{1}{2}\left(u^2\right)_{xx} + u_{ttxx} - \varepsilon^2\delta_1\left(u - w\right), \\ w_{tt} - c_1^2 w_{xx} &= \frac{\alpha_1}{2}\left(w^2\right)_{xx} + \beta_1 w_{ttxx} + \varepsilon^2\gamma_1\left(u - w\right) - \varepsilon^2\delta_2\left(w - z\right), \\ z_{tt} - c_2^2 z_{xx} &= \frac{\alpha_2}{2}\left(z^2\right)_{xx} + \beta_2 z_{ttxx} + \varepsilon^2\gamma_2\left(w - z\right). \end{aligned}$$

DERIVING THE DISPERSION RELATION

We want to derive the **linear dispersion relation** for the above system to help us identify interesting behaviours in the solution for varying order coefficients. First we ignore the non-linear terms and so, for the resulting system, we seek solutions of the form

$$u = u_0 e^{ik(x-pt)}, \quad w = w_0 e^{ik(x-pt)}, \quad z = z_0 e^{ik(x-pt)},$$

where k is the **wavenumber** and p is the **phase speed**. Now we substitute into the linearised system, eliminate the exponential terms and then gather like terms to give

$$\begin{aligned} \left(-k^2p^2 + k^2 - k^4p^2 + \varepsilon^2\delta_1\right)u_0 - \varepsilon^2\delta_1 w_0 &= 0, \\ -\varepsilon^2\gamma_1 u_0 + \left(-k^2p^2 + c_1^2k^2 - \beta_1 k^4p^2 + \varepsilon^2\gamma_1 + \varepsilon^2\delta_2\right)w_0 - \varepsilon^2\delta_2 z_0 &= 0, \\ -\varepsilon^2\gamma_2 w_0 + \left(-k^2p^2 + c_2^2k^2 - \beta_2 k^4p^2 + \varepsilon^2\gamma_2\right)z_0 &= 0. \end{aligned}$$

This system can be written in the form $A\overline{u} = \overline{0}$ where $\overline{u} = \begin{pmatrix} u_0 & w_0 & z_0 \end{pmatrix}^{\mathsf{T}}$ and

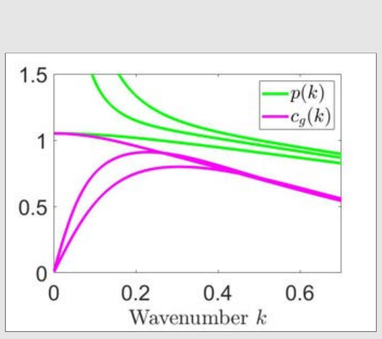
$$A = \begin{pmatrix} k^2\left(1 - p^2\right) - k^4p^2 + \varepsilon^2\delta_1 & -\varepsilon^2\delta_1 & 0 \\ -\varepsilon^2\gamma_1 & k^2\left(c_1^2 - p^2\right) - \beta_1 k^4p^2 + \varepsilon^2\gamma_1 + \varepsilon^2\delta_2 & -\varepsilon^2\delta_2 \\ 0 & -\varepsilon^2\gamma_2 & k^2\left(c_2^2 - p^2\right) - \beta_2 k^4p^2 + \varepsilon^2\gamma_2 \end{pmatrix}.$$

Non-trivial solutions exist when $\det(A) = 0$, and this equation is known as the **linear dispersion relation**. After some manipulation we get the result

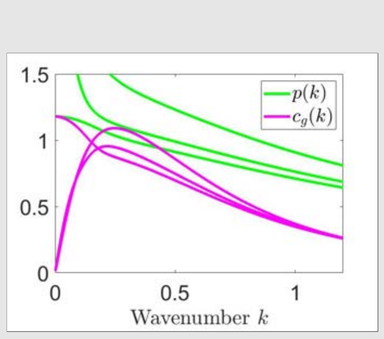
$$\begin{aligned} \left[k^2\left(1 - p^2\right) - k^4p^2 + \varepsilon^2\delta_1\right]\left[k^2\left(c_1^2 - p^2\right) - \beta_1 k^4p^2 + \varepsilon^2\gamma_1 + \varepsilon^2\delta_2\right]\left[k^2\left(c_2^2 - p^2\right) - \beta_2 k^4p^2 + \right. \\ \left. \varepsilon^2\gamma_2\right] - \varepsilon^4\delta_2\gamma_2\left[k^2\left(1 - p^2\right) - k^4p^2 + \varepsilon^2\delta_1\right] - \varepsilon^4\delta_1\gamma_1\left[k^2\left(c_2^2 - p^2\right) - \beta_2 k^4p^2 + \varepsilon^2\gamma_2\right] = 0. \end{aligned}$$

PLOTTING THE DISPERSION RELATION

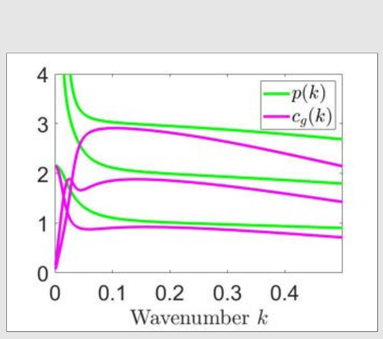
Using MATLAB we can plot p against k as well as the **group speed** c_g, defined as $c_g = p + k\frac{\mathrm{d}p}{\mathrm{d}k}$. We do this for varying orders of magnitude of $(c-1)$ outlined below.



① $(c-1) = \mathrm{O}(\varepsilon)$



② $(c-1) = \mathrm{O}(4\varepsilon)$



③ $(c-1) = \mathrm{O}(1)$

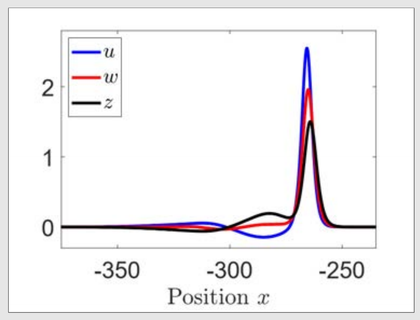
NUMERICAL ANALYSIS

For the **symmetric case** when $c_1 = c_2 = \alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 1$, we let $u = w = z$ which results in a reduced equation (after scaling) for which the travelling-wave solution is known

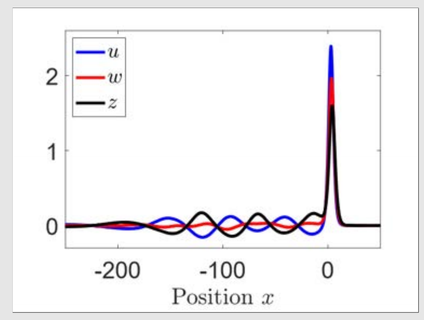
$$u_{tt} - u_{xx} = \varepsilon\left(\frac{1}{2}\left(u^2\right)_{xx} + u_{ttxx}\right) \;\Rightarrow\; u = \frac{3\left(v^2 - c^2\right)}{\alpha}\operatorname{sech}^2\left(\frac{\sqrt{v^2 - c^2}}{2v\sqrt{\beta}}\left(x - vt\right)\right).$$

This solution is known as a **soliton**. We take the initial condition for our system to be the appropriate soliton in each layer, using the coefficients of the respective layer [3].

We use the **pseudo-spectral** numerical method to solve the equations. This consists of taking a Fourier transform to obtain a system of ODEs rather than PDEs. This method prescribes **periodic boundary conditions**. The advantage of using this technique is that it yields high accuracy results and is computationally cheap compared to other methods.

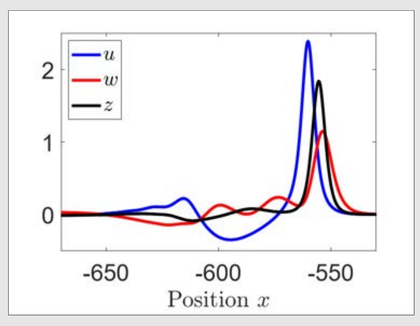


$t = 100$

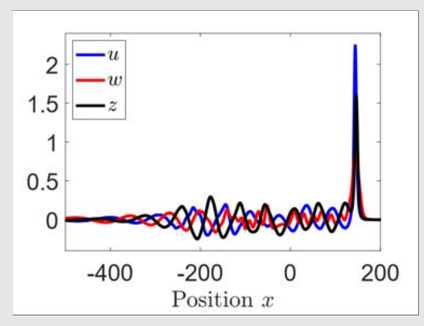


$t = 300$

① $(c-1) = \mathrm{O}(\varepsilon)$ with coefficients $\varepsilon = 0.1$, $c = \alpha = \beta = \left[1, 1 + \frac{\varepsilon}{2}, 1 + \varepsilon\right]$, $\delta = [1, 1, 0]$ and $\gamma = [0, 1, 1]$.

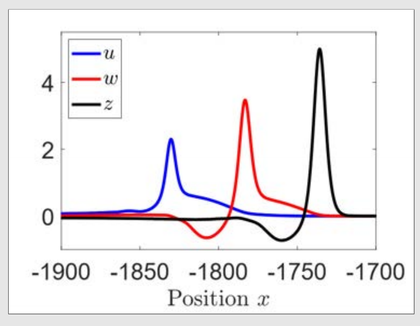


$t = 100$

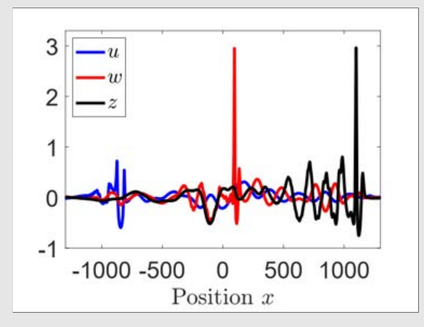


$t = 600$

② $(c-1) = \mathrm{O}(4\varepsilon)$ with coefficients $\varepsilon = 0.1$, $c = \alpha = \beta = [1, 1 + 4\varepsilon, 1 + \varepsilon]$, $\delta = [1, 1, 0]$ and $\gamma = [0, 1, 1]$.



$t = 50$



$t = 1,000$

③ $(c-1) = \mathrm{O}(1)$ with coefficients $\varepsilon = 0.05$, $c = [1, 2, 3]$, $\alpha = \beta = [1, 1, 1]$, $\delta = [1, 1, 0]$ and $\gamma = [0, 1, 1]$.

The first two cases exhibit **radiating solitary waves** and the last case exhibits **Ostrovsky wave packets**.

CONCLUSION & FURTHER WORK

Our results are as we may expect based on similar cases for the **two-layer system**. The instability of the tail in case ② suggests the coupling is about to break which could be a borderline case between ① and ③. More research is needed into what features of the dispersion relation correspond to this change in behaviour of the solution.

References

[1] M. Brocchini, *A reasoned overview on Boussinesq-type models: the interplay between physics, mathematics and numerics*, Proc R Soc A **469** 20130496 (2013).

[2] K.R. Khusnutdinova, M.R. Tranter, *On radiating solitary waves in bi-layers with delamination and coupled Ostrovsky equations*, Chaos **27** 013112 (2017).

[3] K.R. Khusnutdinova, K.R. Moore, *Initial-value problem for coupled Boussinesq equations and a hierarchy of Ostrovsky equations*, Wave Motion **48** 738-752 (2011).

Statistical Energy Analysis in Vibroacoustics

Devika Varsani **Supervisor: David Chappell**

Department of Physics & Mathematics
Nottingham Trent University

Introduction

Statistical Energy Analysis (SEA) dates back to the 1960s, during which it was used by aerospace engineers to predict vibrational response to rocket noise of satellite launch air crafts [1].

SEA incorporates results from fluctuation theory applied to resonators to calculate noise spectra [2]. Applications of SEA are vast, commonly used in ships and cars, it can be used for designing and interpreting experiments to understand vibrational issues in a structure [3].

The aim is to use an SEA model to predict change in vibrational behaviour when the structure is changed. For this the energies need to be known. Once an equation is derived, predictions can be made using a matrix system.

The following assumptions are made in standard SEA modelling:

- Rate of energy dissipation by subsystem i is proportional to the energy E_i.
- Rate of power flow from subsystem i to subsystem j is proportional to the difference in their energies.
- Driving forces on different subsystems are statistically independent - the energy responses can be added to obtain the total mean modal energy.

Why use SEA?

- SEA proves to trump individual mode analysis due to frequency range rising through mode series of structure, results are easier to interpret and less parameters are used in calculations, hence less complicated.
- The use of SEA is more successful as resonance frequency spacing decreases due to an increased number of modes to average, thus a more reliable estimate of the vibration. Hence, SEA is only valid for high frequency vibrations.
- SEA only involves inverting a small matrix as opposed to a Monte Carlo model involving complicated systems of differential equations, making it easier to repeat on a large scale.

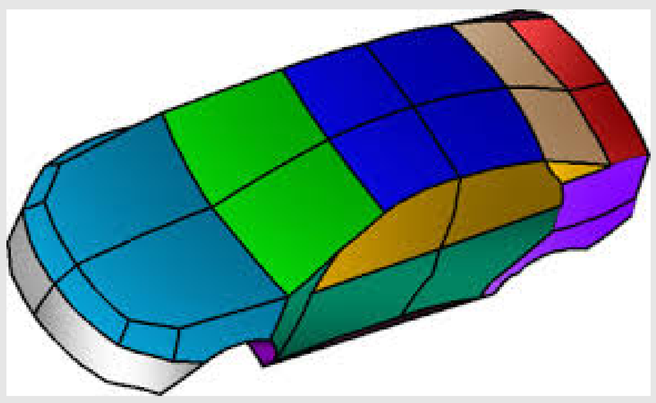


Figure 1: A common application is the vibrational energy of a car body, where each coloured block represents an individual subsystem for which the energy can be found using SEA [4].

Thermal Analogy

The behaviour of energy in high frequency vibrations and thermal form is identical, therefore a thermal analogy can be used to easily understand the structure of vibrations.

In thermal diffusion, there are two elements, one supplied with an external heat source, uniform temperature and each can lose heat via radiation and transfer of energy to the other elements. The interest here lies in the equilibrium temperature under different conditions.

Similarly, consider two coupled substructures, e.g: two plates separated by a beam, then the interest is in the average over the modes and to find the mean modal energy in each substructure within the frequency range as the 'temperature'.

The comparative parameters are:

Thermal diffusion model	**SEA model**
Thermal capacity of element	Modal density
Radiative loss	Damping of vibration modes in the range
Conductivity	Measure of strength of mechanical coupling of substructures

Modal approach to SEA

The ODE system for a pair of coupled oscillators may be written as:

$$m_1\ddot{x}_1 + M(\ddot{x}_1 + \ddot{x}_2) + c_1\ddot{x}_1 + G\ddot{x}_2 + k_1x_1 + K(x_1 - x_2) = f_1 \quad (1)$$

$$m_2\ddot{x}_2 + M(\ddot{x}_1 + \ddot{x}_2) + c_2\ddot{x}_2 - G\ddot{x}_1 + k_2x_2 + K(x_2 - x_1) = f_2 \quad (2)$$

which can be solved using Runge-Kutta methods for example.

We compare the repeated Monte Carlo simulation of this ODE system over a long time duration, where f_1 and f_2 are randomly generated by Gaussian white noise, to the prediction of the mean energy levels given by solving the SEA system:

$$\begin{pmatrix} \beta_{1k} + \eta_1\omega_1 & & -\beta_{ji} \\ & \cdots & \\ -\beta_{ji} & & \beta_{Nk} + \eta_N\omega_N \end{pmatrix} \begin{pmatrix} \langle E_1 \rangle \\ \vdots \\ \langle E_N \rangle \end{pmatrix} = \begin{pmatrix} \langle P_1 \rangle \\ \vdots \\ \langle P_N \rangle \end{pmatrix}$$

with $N = 2$.

- In order for the SEA system to be accurate, the following parameter assumptions must hold true: $M \ll m_1, m_2; K \ll k_1, k_2; G \ll c_1, c_2$ for a system of N coupled oscillators.
- β_{ij} and $\eta_i\omega_i$ can be calculated by use of formulae in [2], pages 73–74.
- The average power $\langle P_i \rangle$ for $i = 1, 2$, is calculated from the input force f_i via the power spectral density [2].

Visualisation

- Due to the random generation of the values for f_i, the mean energy fluctuates, so every time the code is run, a different set of mean energies will be produced.
- The coupled oscillator ODEs are solved using the Runge-Kutta method and hence Monte-Carlo samples of the mean energy over a time period of 500s are presented.
- We compare the mean of the 1000 Monte-Carlo samples with the SEA prediction.

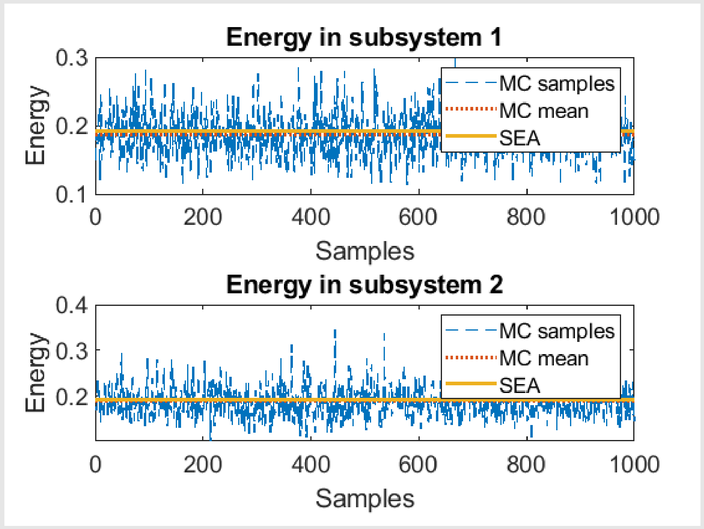


Figure 2: A test case, plots the graphs when initial parameters are equal as: $m_1 = 1, m_2 = 1, M = 0, k_1 = 1, k_2 = 1, K = 0.05, c_1 = 0.1, c_2 = 0.1, G = 0.001$

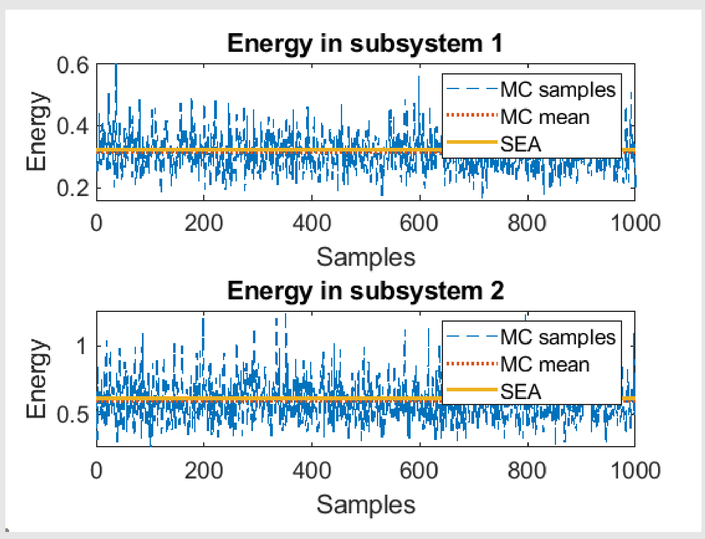


Figure 3: Displays plots for initial parameters: $m_1 = 1, m_2 = 1, M = 0, k_1 = 1, k_2 = 0.5, K = 0.05, c_1 = 0.1, c_2 = 0.05, G = 0.001$

References

[1] F. Fahy. *Philosophical Transactions: Physical Sciences and Engineering*. Royal Society, 1994.

[2] A.Le Bot. *Foundations of Statistical Energy Analysis in Vibroacoustics*. Oxford University Press, 2015.

[3] J Woodhouse. *An Introduction to Statistical Energy Analysis of Structural Vibration*. Topexpress Ltd, 1981.

[4] X Liu and Q Feng. *Automotive Tire Noise and Vibrations*. Butterworth-Heinemann, 2020.

School of Science and Technology

Using Artificial Neural Networks to Detect Brain Tumours

Kendall Feeny
Supervisor: Dr. Archontis Giannakidis
Department of Physics & Mathematics, Nottingham Trent University

NEURAL NETWORKS

Neural Networks are a variant of machine learning algorithms that aim to artificially mimic the operations which are found in our brains:

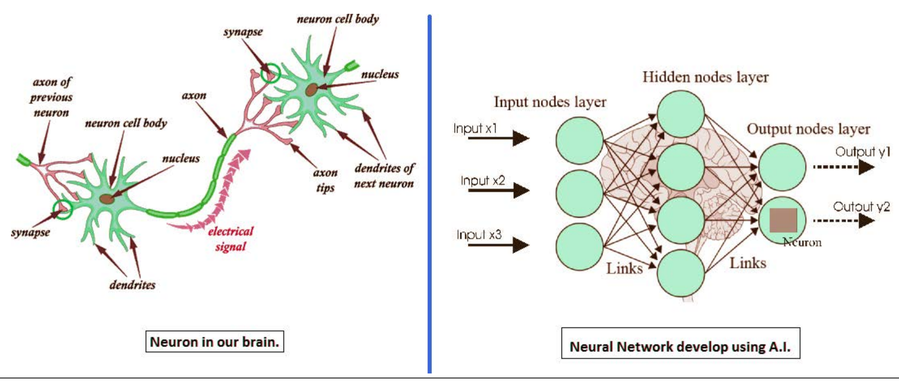


Fig 1: Comparison of a neuron in a human brain and an artificial neural network[1]

At a fundamental level there are two main components: nodes (in place of neurons) and their connections, known as 'edges' (in the place of synapses). Each edge can transmit a signal to multiple nodes. The nodes will process this signal and may pass a further signal onto other connected nodes.

All inputs are assigned a 'weight' which controls the relative impact of each input towards the output prediction. This is combined with an unknown bias term in a stage known as summation to produce a prediction.

Neural networks consist of three components (as shown in the diagram above), the first layer is input data provided to the model and the third layer provides an output prediction. The middle layer is known as the 'hidden layer', this may contain any number of layers, layers can be crossed numerous times before reaching the output layer.

In the hidden layers, several non-linear transformation of the input data occur, combine with a bias term and undergo an activation function to provide a prediction for the output label. These layers are known as hidden because the inner workings are not visible to the system.

PROJECT AIM

The purpose of this project was to, using a data set of Brain MRI Images[2], create a neural network model which correctly identified the images as one of two classes – those with tumours and those without. This classification (with two possible outcomes) is known as binary classification – does the image contain a tumour (1) or not contain a tumour (0).

The images came from an unknown source, with 155 images of MRIs with tumours and 98 images with no tumours – this means that a model which guesses positively 100% of the time would get 61.3% accuracy. This is important to consider when building as the unbalanced dataset may affect the final model. The assessment metric (measure by which we assess the performance of the model) for the neural network will be the accuracy of the model:

$$Accuracy = \frac{Correct\ predictions}{Total\ predictions} \times 100$$

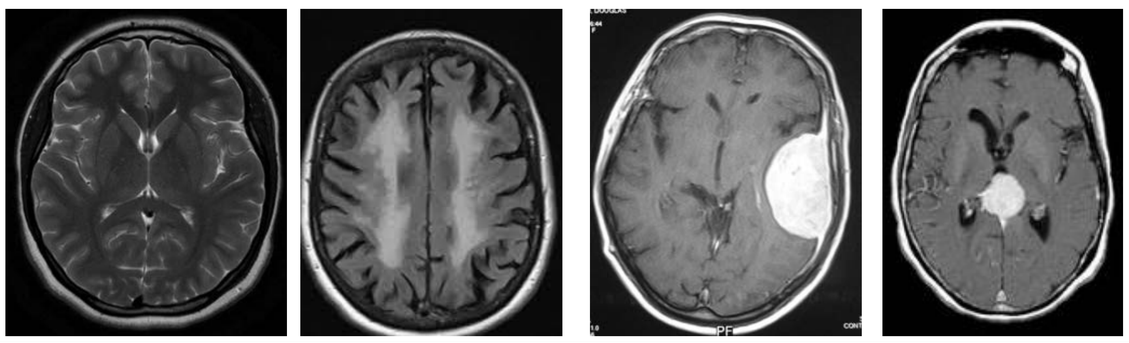


Fig 2 & 3 :(left) two of the MRI scans from the data set with no tumours
Fig 4 & 5 :(right) two of the MRI scans from the data set with tumours

CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks, also known as ConvNets or CNNs are a subset of artificial neural networks used where the input data takes the form of images, speech or audio signals. Our MRI images are made up from a huge matrix of pixels. For some simpler image processing, inputs can be simplified into a vector form. However, in this case the location of neighbouring pixels can be incredibly important and so vectorisation is not appropriate due to oversimplification.[3]

CONVOLUTIONAL NEURAL NETWORKS (CONTINUED)

For this instance, the input image vector has several unique filters applied in order to create many feature maps to summarize the features detected in the imagery. These feature maps, or convolution layers undergo a stage known as 'pooling' in which the layers are adjusted minutely to prevent invariance. All of these inputs are flattened before being passed through a typical neural network.

IMPLEMENTATION

With all machine learning models, there are a series of parameters which impact the accuracy, effectiveness and cost of a model. Through adjustment of these parameters, we can produce a model which performs well whilst balancing the cost.

There are 23 different parameters which can be adjusted using Sklearn. However, a search of all of these would be very exhaustive and inefficient. So instead, I researched and selected 3 features which have a significant impact of model accuracy:
- Iterations
- Hidden layer size
- Learning Rate

Each feature will have varied impact on the accuracy individually, but there can be a significant improvement when ideal combinations of each feature are utilised. To search the feature combinations exhaustively, I utilised a built in function called *GridSearchCV*.

This forms a table with X dimensions (where X is the number of features) and will create a version of the model utilising each possibly combinations of the variables – as the production of this many models is time exhaustive, I could only use a restricted range of potential options. Utilising this, I then produced a final optimised model and predicted unseen labels for all the test set data to test the performance of this model.

RESULTS

Utilising the below accuracy calculation equation, I calculated the model accuracy for predicting the testing data labels to be 87.76% to 2 decimal places. This indicates that the model is performing well and is very positive.

Through viewing the individual mistaken predictions, many false positives include larger areas of white which could plausibly be mistaken for tumours – something which would probably improve in future versions.

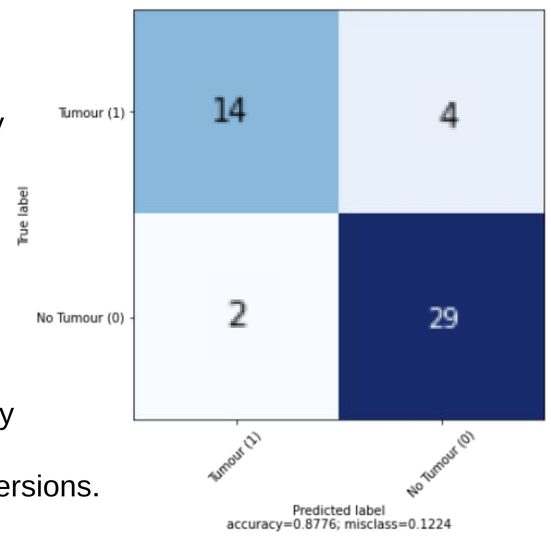


Fig 6: A confusion matrix showing the prediction results of the convolution neural network with tuned parameters

$$Accuracy = \frac{Correct\ predictions}{Total\ predictions} \times 100$$

$$Accuracy = \frac{43}{49} \times 100 = 87.76\%$$

FUTURE WORK

Given more time, and additional computing power, I feel that I could improve the model produced during this internship. Although an accuracy of 87.76% is good, there is the opportunity to improve the model for future work.

The most obvious limiting factor to the development of this model was the insubstantial available data. The provision of subsequent larger datasets will allow for the model to process more types of tumour – as the tumour location, size and colour varies from image to image. Or alternatively, in the future I could utilise data augmentation to "increase" the sample size.

Alternatively, narrowing the types of tumour that are being detected would improve accuracy as it would decrease variation in imagery – for instance, only looking at frontal lobe tumours.

Experimentation is also something that was restricted due to time in this project, as stated above there were 20 parameters which I did not have the opportunity to adjust within the model – this offers the possibility for future development and expansion.

[1] = Data Flair, 2018. Artificial Intelligence vs Machine Learning vs Deep Learning vs Data Science. [Online] Available at: https://data-flair.training/blogs/artificial-intelligence-vs-machine-learning-vs-dl-vs-ds/ [Accessed September 2021]

[2] = Chakrabarty, N., 2019. Brain MRI Images for Brain Tumour Detection. Kaggle Data. [Online] Available at: https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection [Accessed August 2021]

[3] = Rocco, I., Arandjelovic, R. & Sivic, J., 2017. Convolutional neural network architecture for geometric matching. CVPR - IEEE Xplore, Volume IV, pp. 6148-6157.

[4] = Zhang, C., Zhou, P., Li, C. & Liu, L., 2015. A Convolutional Neural Network for Leaves Recognition Using Data Augmentation. IEEE International Conference on Computer and Information Technology, pp. 2143-2150.

NOTTINGHAM TRENT UNIVERSITY

Numerical Analysis of Shallow Water Equations

Matthew Cuthbertson N0785362

Department of Physics & Mathematics
Nottingham Trent University

INTRODUCTION

The **Navier-Stokes equations** are some of the hardest equations to solve both analytically and computationally; so much so that proving existence and smoothness of solutions is one of the millennium problems, offering a $1 million prize if solved [1]. They are fundamental to understanding fluid dynamics as a whole.

The Navier-Stokes equations are mathematical statements for the **conservation of momentum**, and the **conservation of mass**, within Newtonian fluids.

The **shallow water equations** are derived from these, and the equations under analysis are

$$\frac{\partial \omega}{\partial t} + [\psi, \omega] = v\nabla^2\omega \tag{1}$$

$$\nabla^2\psi = \omega, \tag{2}$$

where

$$[\psi, \omega] = \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} - \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x}.$$

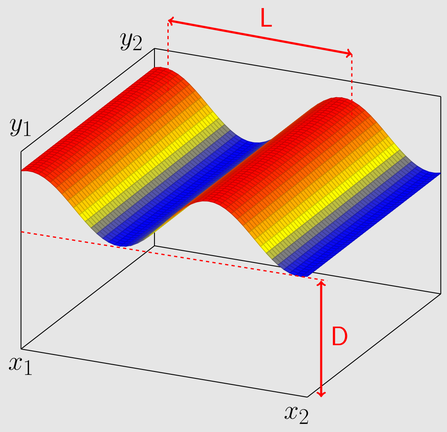
Here,

- ψ denotes the streamfunction,
- ω denotes the fluid vorticity, and
- ν denotes the fluid viscosity.

Reference [1]: Clay Mathematics Institute, n.d. Available at: `https://www.claymath.org/millennium-problems/navier-stokes-equation`

DERIVATION

The applications of shallow water equations are that they can be used to model waves in lakes, rivers, and even the atmosphere and gravitational waves. For them to work, one of the main assumptions when modelling phenomena is that **the wavelength affecting the fluid must be much larger than the depth of the basin containing the fluid**. Let us assume we have a fluid bounded between $x \in [x_1, x_2]$ and $y \in [y_1, y_2]$. The below diagram shows a basic representation of this.



Here, we require that $\delta = D/L << 1$. We also note that the system has the following properties:

- some velocity field $\vec{v} = \begin{pmatrix} u & v & w \end{pmatrix}^T$, and
- some vorticity field $\vec{\Omega} = \begin{pmatrix} \omega_x & \omega_y & \omega_z \end{pmatrix}^T$.

Since we are only interested in the vorticity in the $x - y$ plane, we take $\omega = \omega_z$. Since $\omega = \nabla \times \vec{u}$, we see that $\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$.

Conservation of Mass

The mass of the system at any time t is defined to be

$$M = \int_{x_1}^{x_2}\int_{y_1}^{y_2} \rho(x,y)h(x,y,t)\ \mathrm{d}x\mathrm{d}y, \tag{3}$$

where ρ is the density of the fluid at any point in space, and h is the height above the base at time t at any point in space. Taking the partial derivative across equation (3) with respect to time yields the rate of change of mass across the whole system, the flux in the x direction, and the flux in the y direction. Applying the fundamental theorem of calculus and noting that mass must be conserved yields the following result:

$$\frac{\partial h}{\partial t} + \frac{\partial (uh)}{\partial x} + \frac{\partial (vh)}{\partial y} = 0 \implies \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{4}$$

assuming h to be constant.

Conservation of Momentum

The two resulting equations are derived directly from the Navier-Stokes equations, and are as below:

$$\frac{\partial (uh)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial (huv)}{\partial y} = fhv, \tag{5}$$

$$\frac{\partial (vh)}{\partial t} + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) + \frac{\partial (huv)}{\partial x} = -fhu. \tag{6}$$

Bringing these together...

Differentiating equations (5) and (6) with respect to t, subtracting (6) from (5) and utilising previous definitions, we arrive at the result

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = 0. \tag{7}$$

By introducing the definition of the streamfunction ψ as

$$u = -\frac{\partial \psi}{\partial y}; \qquad v = \frac{\partial \psi}{\partial x}, \tag{8}$$

we arrive at the desired result in (1). By also differentiating u and v with respect to t, we arrive at the result shown in equation (2).

What do these equations represent?

Equation (1) is known as the *advection-diffusion* equation and describes how waves move with relation to vorticity and how they dissipate over time. Equation (2) describes the streamfunction in terms of the vorticity, and this ensures that the fluid is incompressible, i.e. $\boldsymbol{\nabla} \cdot F = 0$ for all $x,\ y$.

FINITE DIFFERENCE METHODS

Here, we require the discretisation of equations (1) and (2). This will involve deriving numerical schemes with the aim of simulating various situations using MATLAB. This can be done by time-stepping using Euler's method, and space-stepping using central difference approximations. To begin, we use equation (2) to define an initial streamfunction $\psi_0(x, y)$ given an initial vorticity $\omega_0(x, y)$. Then, we advance in time by using equation (1).

Discretisation

To fully discretise the system of equations, we consider an $M \times N$ meshgrid. We need to use the formulae for the central difference approximation on the Laplacian $\nabla^2\psi = \omega$, which is an elliptic PDE. This expands to $\psi_{xx} + \psi_{yy} = \omega$. This leads to the following central difference:

$$\frac{\partial^2 \psi}{\partial x^2} \approx \frac{\psi(x + \Delta x, y, t) - 2\psi(x, y, t) + \psi(x - \Delta x, y, t)}{\Delta x^2}. \tag{9}$$

The formula is very similar in the case of ψ_{yy}. This reduces to

$$\psi_{(m-1)n} + \psi_{m(n-1)} - 4\psi_{mn} + \psi_{(m+1)n} + \psi_{m(n+1)} = \delta^2\omega_{mn}. \tag{10}$$

At the boundaries, we also have that $\psi_{1n} = \psi_{(M+1)n}$ and $\psi_{m1} = \psi_{m(N+1)}$. From this, we can derive a single block matrix to represent the Laplacian as an operation. We also must derive matrices for the first derivatives in equation (1). These matrices are then plugged back in to yield

$$\frac{\partial \omega}{\partial t} = \nu A\vec{\omega} + B\vec{\psi} \cdot C\vec{\omega} - C\vec{\psi} \cdot B\vec{\omega}. \tag{11}$$

MATLAB Implementation

The above was scripted into MATLAB to run simulations on a 128×128 grid with initial vorticity $\omega_0(x, y) = \exp\left(-2x^2 - y^2/20\right)$ and viscosity $\nu = 0.001$. Simulations are shown below:

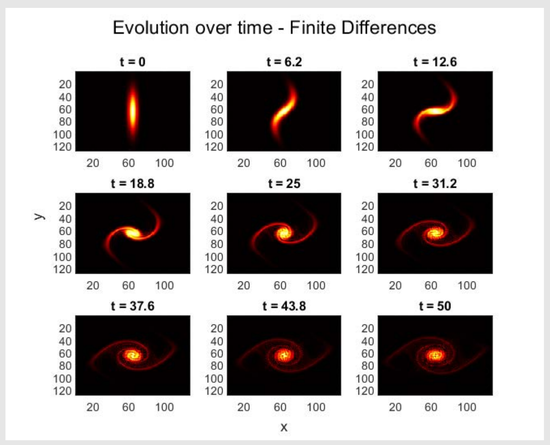


Fig. 1: 50-second evolution

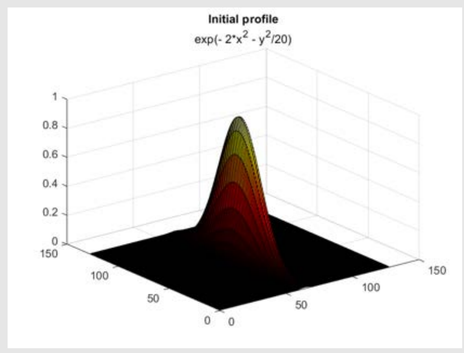


Fig. 2: Initial profile

SPECTRAL METHODS

For this method, we make use of the Fast Fourier Transform algorithm ("FFT"). This is a method that requires knowledge of more advanced mathematics, but it is **a lot quicker** computationally than the aforementioned finite difference method. Here, we begin similarly by generating the same $M \times N$ meshgrid, and then taking the Fourier transform of an initial profile ω_0. Using MATLAB, we this initial profile ω_0 is already represented by the meshgrid from before, so the `fft2()` command is run directly on it. This is then reshaped into a single column vector to be passed into `ode45()` to derive a solution.

MATLAB Implementation

Using the initial profile $\omega_0(x, y) = \exp\left(-2x^2 - y^2/20\right)$ and viscosity $\nu = 0.001$, the following simulation was run:

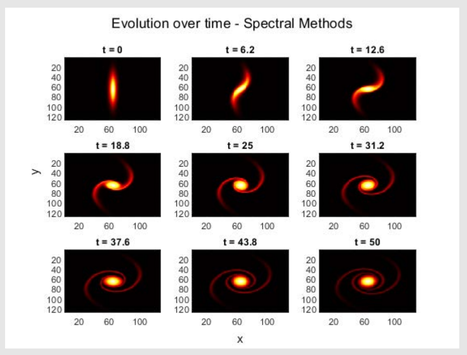


Fig. 3: 50-second evolution

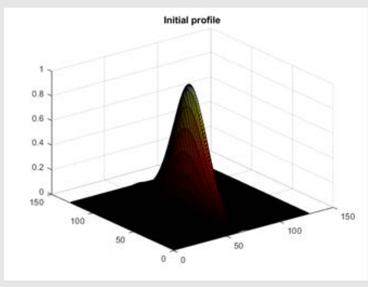


Fig. 4: Initial profile

Notice that these are very similar to the finite difference simulations shown above. In these spectral simulations, the 'arms' of the waves at $t = 37.6$ onwards are more crisp, which represents the higher accuracy of this spectral method.

KEY FINDINGS AND FUTURE WORK

The key findings from this study were that although both spectral and finite difference methods work well for solving the shallow-water equations, spectral methods are much much faster computationally, as shown:

Description	FDM time (s)	SPEC time (s)
Single Gaussian	375.34	2.14
Double Gaussian	963.29	4.46
Triple Gaussian	>1,200	9.69
Alternate Gaussians	>1,200	4.80

We can safely conclude that out of the two methods shown above, Spectral methods are better overall, both computationally and with regards to accuracy. Future work includes the use of another method, Chebyshev polynomials, to solve the shallow water equations, and to apply their use to real-world scenarios e.g. storms, atmospheric movement, etc. and possibly evaluate their use/accuracy against other modelling equations i.e. Boussinesq, Korteweg-de Vries, etc.

References

Kutz, J.N., 2013. *Data-driven modeling & scientific computation: methods for complex systems & big data.* Oxford University Press.

Trefethen, L., 2000. *Spectral methods in MATLAB.* Society for Industrial and Applied Mathematics.

Synchrony in Neural Networks

Ryan Shields

Department of Physics & Mathematics
Nottingham Trent University

INTRODUCTION

Whilst modelling brain networks, it is vital to grasp the dynamics of our networks analytically and visually. We set out to research and identify the nature of synchronisation by utilising ordinary differential equation models of coupled oscillators using data from various research articles. By utilising the Kuramoto Model to describe neural activity, we investigated the impact of altering model parameters on synchrony properties of cortical networks to provide a more tractable model to capture the phenomena of synchrony.

The Kuramoto Model

Firstly, we want to look at the Kuramoto model acting on a network as follows:

$$\frac{d\theta_j}{dt} = \omega_j + S\sum_{k=1}^{N} A_{jk}\sin(\theta_k(t) - \theta_j(t) - \beta),\ \ j = 1,2,...,N, \tag{1}$$

Where θ_j is the phase of the jth oscillator at time t; ω_j is the intrinsic frequency of oscillator j; N is the total number of oscillators; β is the phase offset and A_{jk} is the coupling from oscillator k to j. By utilising the measure of synchrony denoted by R:

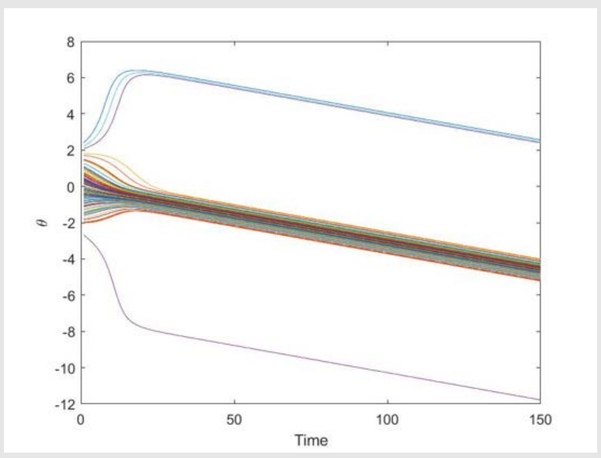
$$Re^{i\theta} = \frac{1}{N}\sum_{j=1}^{N} e^{i\theta_k}, \tag{2}$$

We can in turn simplify model 1 to the following:

$$\frac{d\theta_i}{dt} = \omega_j + RSn_j\sin(\theta_k(t) - \theta_j(t) - \beta). \tag{3}$$

100 oscillator Kuramoto Model for a complete network

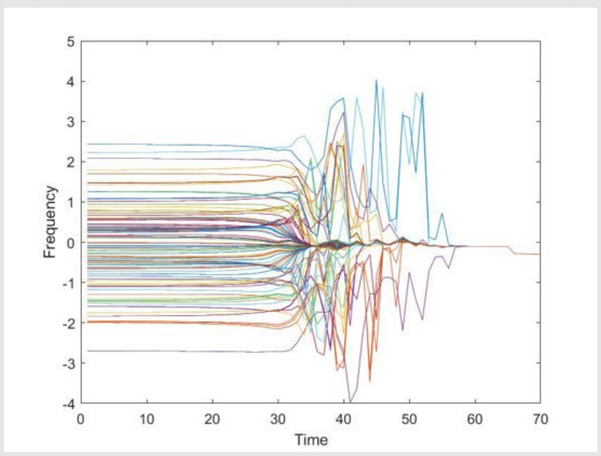
Considering oscillators of $N = 100$ with intrinsic frequency's generated at random through a normal distribution in MATLAB, we must integrate our simplified model 3 and hence plot the result across time producing the following illustration:



We notice that many of our points converge into a synchronised state as time increases shown through our frequencies moving towards one another. However, to define this stationary state that our points converge to, we may utilise ϕ to determine a phase locked solution. The remaining oscillators will be rotating out of synchrony alongside them. Thus, ϕ defines a constant solution within the model and is illustrated as follows:

$$\phi_j = \sin^{-1}\left(\frac{\triangle_j}{Sn_jr_j}\right) + \Phi,\ \text{for j} = 1,2,...,N. \tag{4}$$

This derivation was possible through knowing the difference between our natural frequencies $\triangle_j \equiv \omega_j - \Omega$, where Ω is our population parameter. Secondly, knowing $\phi_j = \theta_j - \Omega t$. Finally, to define a fixed point within our model, we may set $\frac{\partial\phi_j}{dt} = 0$ and permit $\Phi = \theta_j - \Omega t$. Thus, using the knowledge gathered, we return equation 4 and can plot the result over time.

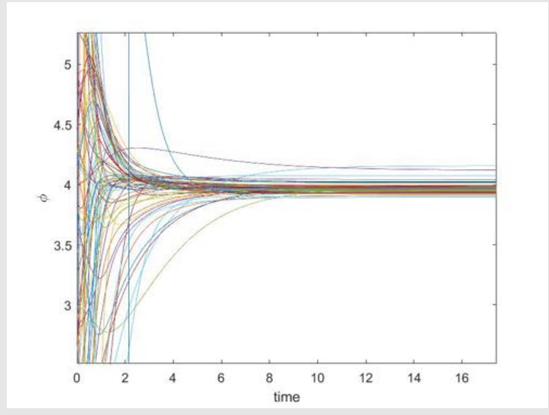


Notably, our widespread frequencies all tend to a fixed solution of 0 in this case; indicating synchronisation towards a stationary state. This is simply demonstrated graphically by our frequencies converging to a frequency of 3 as time increases.

Macaque cortical connectivity for 71 nodes

The dynamics of our nodes represent coupled oscillators and the edges between connect to form a complete network of direct links alongside a phase offset of 0. The cortical connectivity within the macaque brain consists of 71 nodes of random frequencies. The connectivity tends towards a phase locked solution. Thus, given a strong enough coupling strength, our model can asymptotically approach a stable solution obtained via the following:

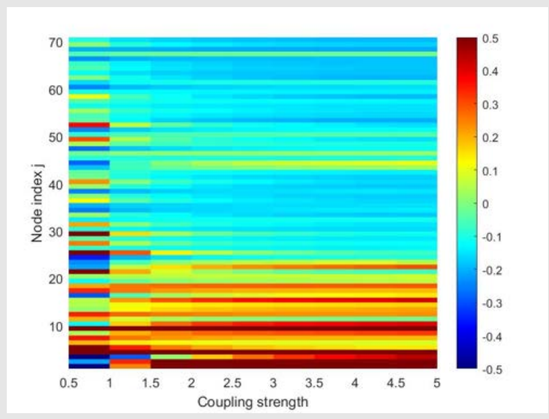
$$\phi_j = \sin^{-1}\left(\frac{\triangle_j}{Sn_jr_j}\right) + \Phi - \beta,\ \text{for j} = 1,2,...,N \tag{5}$$



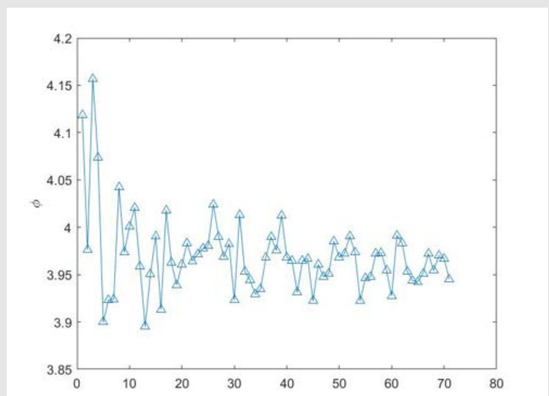
Visually, we can see that our phase ϕ tends to a constant solution. Thus, we can deduce that the oscillators either completely synchronise or phase lock. Here, as we can see very few have not completely synchronised, we infer that the following solution is phase locked.

Network degree for phase lag/lead in neural networks

Finally, we can present the simulation results of our macaque brain data as follows:



From the plot, we are visualising the phase values of our 71 nodes as our coupling strength increases over time. By aligning our node index in ascending order, we can visualise the phases by a distinct heat map. We note that the blue colour infers that the nodes of our macaque data are phase lagging, and the red colour denotes that the nodes are phase leading. Furthermore, we can analytically depict our phases being maintained whilst our coupling strength increases over time as follows:



Evidently, given a node within our macaque brain with a large enough degree, our nodes begins to experience phase lagging.

References

1 : Moon, J.-Y., Kim, J., Ko, T.-W., Kim, M., Iturria-Medina, Y., Choi, J.-H., Lee, J., Mashour, G.A. and Lee, U. (2017). *: **Structure Shapes Dynamics and Directionality in Diverse Brain Networks*** Mathematical Principles and Empirical Confirmation in Three Species. Scientific Reports, [Accessed 10/06/2021].

2 : Sue Ann Campbell (2007) *: **Time Delays in Neural Systems.*** Handbook of brain connectivity. Springer, Berlin, Heidelberg, [Accessed 10/06/2021].

The Maths of Condensed Matter Physics

Sam Holmes and Ryan Shields **Supervisor: Dr. Mark Wilkinson**

Department of Physics & Mathematics
Nottingham Trent University

What Are Liquid Crystals?

Liquid crystals are an intermediary state between crystalline solids and isotropic liquids. Introducing an electromagnetic field to these phases has created a multi-billion pound industry all based on liquid crystal displays (LCDs), which are used in billions of phones, laptops, PCs, TVs and watches all around the world. The characteristics of liquid crystals come from the shape, orientation, position, and other properties of their molecules and how they interact. The most common thermotropic liquid crystals have liquid crystalline phases existing at a temperature above that of an isotropic fluid but below that of a solid. The phases of liquid crystals are determined by the order of orientation of their constituent molecules. The liquid crystals used in industry are usually comprised of a mix of different types of molecules to increase performance. Examples of these thermotropic liquid crystals are MBBA and 5CB, which consist of molecules with lengths of 2-3nm.

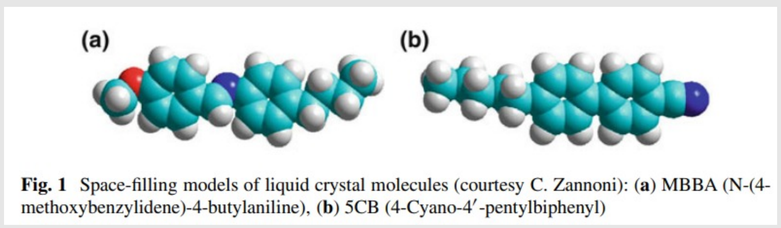


Fig. 1 Space-filling models of liquid crystal molecules (courtesy C. Zannoni): **(a)** MBBA (N-(4-methoxybenzylidene)-4-butylaniline), **(b)** 5CB (4-Cyano-4′-pentylbiphenyl)

Here, these molecules that make up a small blob of a liquid crystal are approximately shaped like rods and can be modelled using ellipsoids of revolution. There are three main phases of liquid crystals: nematics, smectics and cholesterics. The only phase we are considering here are nematics. The molecules in the nematic phase have orientational order but no positional order. This means we can define a mean orientation of these molecules about the long axis by a unit vector n = n(x,t), at point x and time t, called the director. Smectic molecules have orientational and some positional order. Cholesteric molecules have a helical structure with either the long or short axis perpendicular to the director n.

The temperature that the nematic phase exists at is normally achieved through cooling from a critical temperature. An isotropic liquid is cooled down until the temperature is below that of the critical temperature, meaning that the molecules go from having no orientational order to some orientational order. For $\theta > \theta_c$ the material is isotropic, for $\theta_m < \theta < \theta_c$ the material is in the nematic phase and finally $\theta < \theta_m$ the material could be a different liquid crystal phase or simply a solid phase. For example MBBA has critical temperatures $\theta_m \approx 17, \theta_c \approx 45$.

The Order Parameter Q

In order to model a liquid crystal the use of a continuum description is needed. The necessary variables needed to do this are called order parameters. These describe the nature and amount of order in a liquid crystal. One order parameter that has already been introduceds is the director $n = n(x,t)$, a unit vector which describes the mean orientation of all molecules at a point x at time t. Only static configurations of liquid crystals will be studied here for simplicity, where the velocity of fluid is negligible. Meaning that the continuous variables will solely depend on x. Liquid crystal molecules are modelled using an open bounded region $R \subset \mathbf{R}^3$ normally shaped like a rod or ellipsoid and all approximately have a similar shape and symmetry.

The function $f(\theta)$, measures the probability of orientation of each molecule in a container. By averaging the probabilities over a fixed time period, a function f, which has the following properties will be obtained

- $\int_0^{2\pi} f(\theta)\, d\theta = 1$ (total probability is 1)
- $f(\theta + \pi) = f(\theta)$ (the molecules have head-to-tail symmetry)

If the orientation of molecules are equally distributed in all directions then an isotropic fluid is given by $f(\theta) = \frac{1}{2\pi}$.
Naturally, one could think to use this function f as an order parameter. However, this function has an infinite dimension which is impossible to compute. Therefore, an approximation of f by way of moments is best. The first moment cancels out due to the head-to-tail symmetry and the second moment is a symmetric tensor M given by

$$M := \int_0^{2\pi} f(\theta)(\mathbf{n}(\theta) \otimes \mathbf{n}(\theta))\, d\theta \tag{1}$$

A system with no order, has $f(\theta) = \frac{1}{2\pi}$, such a system is denoted M_0, which is given by $\frac{1}{2}I$. N.B. this is a symmetrical two dimensional matrix with trace 1.

Now, a suitable analogue for M in three dimensions is a matrix with $\frac{1}{3}$ on the leading diagonal. Using M and M_0 the Q-tensor order parameter can be calculated which measures the deviation of M from it's isotropic value M_0:

$$Q := \int_0^{2\pi} f(\theta))(\mathbf{n}(\theta) \otimes \mathbf{n}(\theta) - \frac{1}{3}I)\, d\theta \tag{2}$$

Deriving the minimizer of the Q tensor

So far, we have constructed Landau-de Gennes Energy Functional

$$E[Q] = \int_\Omega \left[\frac{L}{2}|\nabla Q|^2 + \frac{a(\theta)}{2}tr(Q^2) + \frac{c}{4}tr(Q^2)^2\right] dx \tag{3}$$

or by way of Einstein summation notation

$$E[Q] = \int_\Omega \left[\frac{L}{2}\partial_k Q_{ij}\partial_k Q_{ij} + \frac{a(\theta)}{2}Q_{ij}Q_{ji} + \frac{c}{4}(Q_{ij}Q_{ji})^2\right] dx \tag{4}$$

whereby $E : C^1[a,b] \to R$. We wish to derive the minimizer of our energy functional, or the local extrema by calculating the first variation of E $\delta E(u;v)$ and allowing v

$$C_0^1[a,b] = \{v \in C_0^1[a,b] : v(a) = v(b) = 0\} \tag{5}$$

to be of admissible direction so that we may define the necessary geodesics between our points. As a result of the above notion, we find that

$$\frac{d}{dt}E[Q + t\phi]_{t=0} = \int_\Omega L\partial_k\phi_{ij}\partial_k Q_{ij} + a(\theta)tr(Q\phi) + tr(Q^2)tr(Q\phi)c\, dx \tag{6}$$

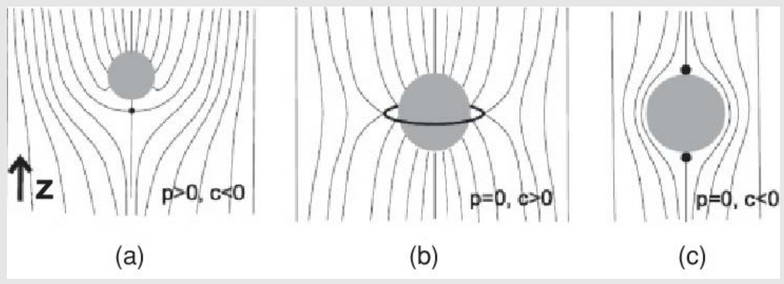
and after further calculation and use of the fundamental lemma of the calculus of variations, we find that

$$-L\nabla^2 Q + aQ + tr(\dot{Q}^2)Qc = 0 \tag{7}$$

defining the necessary PDE for the Q tensor, or the associated Euler-Lagrange equation. We can take this further by claiming the Q tensor $Q_\lambda(x)$ that models defects in a liquid crystal is the solution to the above result, given by:

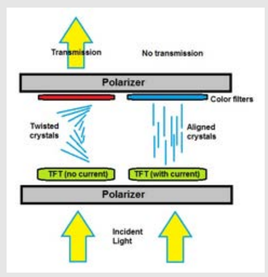
$$Q_\lambda(x) = S(|x|)\left(x \otimes x - \frac{1}{2}|x|^2 I\right) \tag{8}$$

such that $S : [0,\infty] \to R$ is unknown. formally defined as a hedgehog defect, which can then be used to derive an ODE from the above PDE.



Applications of liquid crystals

One of the major applications of liquid crystals is liquid crystals displays (LCDs). The type used in displays are twisted nematic. This type of liquid crystal is sandwiched between two glass plates that have a polarizer attached to them. These polarizers are aligned perpendicular to each other so that the light that goes through the first polarizer can only go through the second one if the light is twisted by the liquid crystal. If the light is twisted correctly then the pixel appears bright. The twisting of the light can be stopped by using an electric field that is perpendicular to the glass plates. These molecules are then aligned parallel to the electric field meaning the light is not twisted and the pixel is dark. A diagram of this can be seen below.



FUTURE WORK & CONCLUSIONS

Given more time, we would have worked through the derivation of the associated ODE

$$-Ls''(r) - \frac{5L}{r}s'(r) + as(r) + \frac{c}{2}s(r)^3 r^4 = 0. \tag{9}$$

This second-order, inhomogeneous, non-linear ode yields a difficult problem to solve. Another approach to consider in terms of future work would be to work in 3-dimensions. Notably, the problem would drastically increase in difficulty and would take longer than the placement offered to consider thoroughly.

References:
Ball, J.M. (2017). Mathematics and liquid crystals. Molecular Crystals and Liquid Crystals, [online] 647(1), pp.1–27. doi:10.1080/15421406.2017.1289425.