Week 1

My first correspondence with Dr Graeme Auckland pertaining to the Pandemic senior honours project was on January 3rd, 2023. The following day we organised a teams meeting and discussed the goals and targets for the project. Here I assigned myself the tasks of completing the numerical integration of the SIR functions in python, reading about potential future models that may fit better, and drawing vector fields to illustrate the differential equations of pandemic modelling.

I spent the following week focusing on what the final goal of the project would be. I came to Graeme with the suggestion of trying to recreate the real UK pandemic infection curve through modelling and data analysis. This was seen as incredibly ambitious as apparently nobody has managed to achieve this before, therefore the goal was adapted to an attempt at recreating the graph and seeing how close I could get.

For this to work the SIR model would not be sufficient for COVID-19. Suffers of COVID-19 tend to have an incubation period of several days where they are symptomless, furthermore, studies suggest that people who have recovered only have temporary immunity, and vaccination efforts were implemented in two doses. Therefore, the modified version of the SEIRS model was used as this best fit the nature of the COVID-19 virus where compartments V1 and V2 were added to include vaccinated individuals with their own coefficients, Birth and death rates were then added allowing for a non-constant population. Of course, many details can be added such as age categories, COVID-19 variants, and migration. However, I consider the model to have enough detail after this. Future goals include implementing time dependant coefficients based on real world data, cleaning up the code so that it is efficient, implementing a grid-based simulation based on the models I have construct, and visualising the numerical integration in different ways to convey new information. The project is going well however I'm still not convinced of my final goal which I'm unsure I'll be able to achieve to a satisfactory level.

Week 2

This week I further refined my Pandemic model which I've now called the SEIRSV2 model. I've also included birth and death rates in the model and drawn many graphs including a stackplot, lineplot, piechart, and vector fields. Furthermore, I've also created an ASCII line and grid spread of the simulations. However, the code is incredibly inefficient which is something that I plan on fixing. I've also become bogged down in the larger model and so I'm considering creating two models. One based on the SIRS and the other based on SEIRSV2 which much more closely relates to the covid-19 pandemic. This way, it is easier to develop, debug, and lessons applied to the SIRS model can then be applied to the SEIRSV2 model. This week I also decided to use Our World in Data as the source for all my data, in particular, the R number, vaccination rates, and infection curves can all be gathered from the same source which can be used for reference when assessing the model and be used to calculate the various coefficients.

To ensure readable code, object-oriented programming was used where the model's coefficients would be stored as attributes and the numerically integrated functions would be calculated through methods. This also allowed for the grouping together of the ASCII models where various methods were used to simulate the grid spread. The ASCII models were constructed by creating a 2D array and updating each element of the 2D array depending on the state of its neighbours much like John Conway's game of life.

Goals for the following week include expanding the SIRS and and graphical models, and gathering sources which should aid the model, along with preparing the first stages of the report.

Week 3

This week I studied various papers published concerning the COVID-19 pandemic to get a starting point of what the coefficients for my model should be like. I often had to manipulate the data to gain coefficients that would work for my model. For example, to get a birth-rate coefficient I took the average number of people born per day in 2020 and divided it by the population by the end of 2020. For simplicity I assumed that this would remain constant regardless of the total size of the population and that multiplying this coefficient by the current population in the simulation would give the number of births. A similar procedure was carried out for the various deathrates. Other coefficients were simply reciprocals of times, such as the recovery time being the reciprocal of 10 since the average time to recovery is roughly 10 days. Below are the sources for all my calculated coefficients.

https://www.statista.com/statistics/281296/uk-population/
https://www.statista.com/statistics/281981/live-births-in-the-united-kingdom-uk/
https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationprojections/bulletins/nationalpopulationprojections/2018based
https://www.cdc.gov/media/releases/2021/p0607-mrna-reduce-risks.html
https://ourworldindata.org/covid-deaths?country=~GBR
https://www.thelancet.com/journals/ebiom/article/PIIS2352-3964(22)00584-9/fulltext
https://www.sciencedirect.com/science/article/pii/S2254887420301466
https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/diagnosed-with-covid-19-what-to-expect
https://www.goodhousekeeping.com/health/a40062545/how-long-does-covid-symptoms-last-vaccinated/

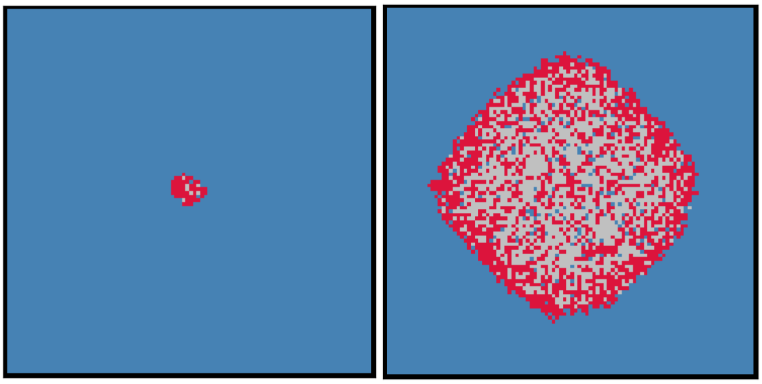
https://ourworldindata.org/covid-deaths?country=~GBR

It was also this week I made long term plans for the rest of project, listing all the major programming tasks. I placed these tasks in a to-do list app to aid in the organisation of the project.

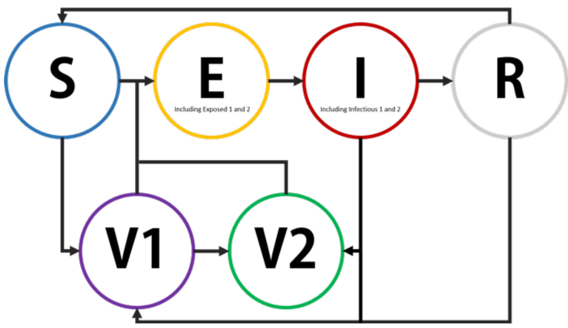
1. Represent grid simulation graphically
2. Increase size of grid simulation
3. Draw vector fields of simulations
4. Calculate new beta values
5. Repeat previous tasks experiment with SIRS model
6. Calculate speed of spread in grid simulations

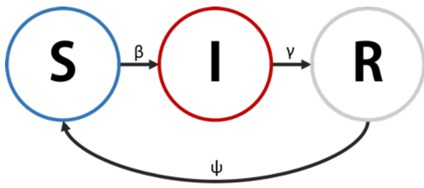
Week 4

Thanks to the temporary scaling back of the project I was able to complete some more complex tasks. I was able to graphically represent the grid simulation with matplotlib. blue represents susceptible individuals, red represents infectious individuals, and grey represents recovered individuals. Following a consistent and conventional colour scheme will minimise confusion for the reader. Rather than numerically integrating equations the total number of infections can be plotted by summing the total number of infected grid squares. However, this relies on random variables therefore it is more stochastic.



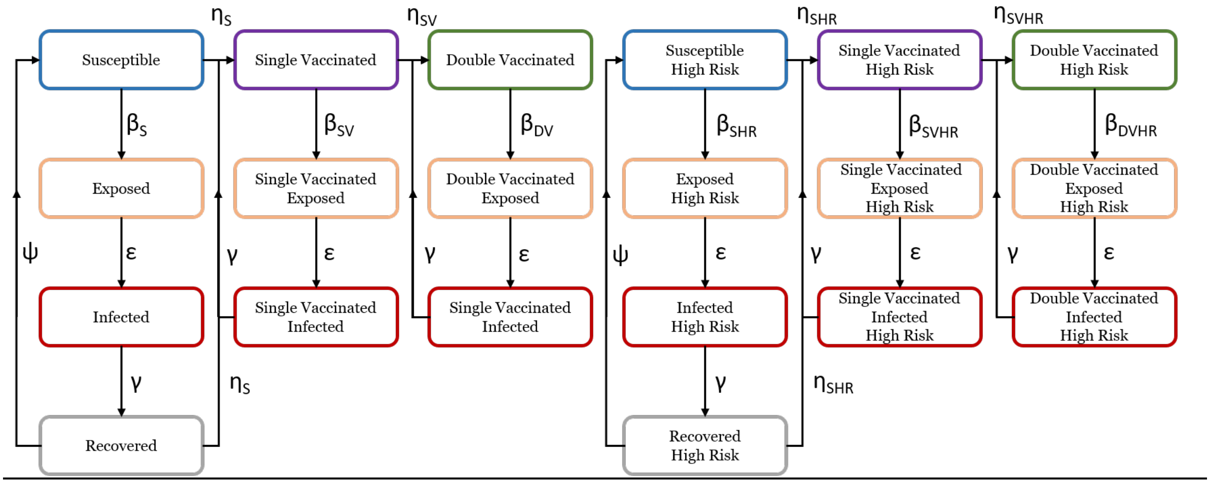
This simplified model can still give insights into the spread of COVID-19 and it will be interesting to see how accurate it is compared to the SEIRSV2. I also drew some diagrams in PowerPoint to show how the models work without having to display the full 6 differential equations.





This week I also began drafting sections of the report after completing an outline of what I want to include in the introduction with the idea being next week I would have at least 5 pages I could choose from to show my supervisor. Beyond this I did some basic cleaning of the code to improve efficiency, implemented a single method to generate vector fields rather than 3, and introduced time dependant coefficients in the model by implementing real world data. In the following week I plan to tune my coefficients so that I can more closely represent what the real infection curve looks like. Furthermore, I plan to use data from deaths to work around the limited testing in the early days of the pandemic.

Week 5



This week was a success for the mathematical introduction section of my report. I want to have a section where I prove the stability of the SIR model and find all infinitly many stable states, after a few hours of research I learned of the Lambert-W function, and how the Stable state could be written in terms of the Lambert-W with some algebraic manipulation. Once I found the correct expression to substitute into the Lambert-W function, I implemented in python function to obtain any stable state. This stable state depends only on the initial conditions of the system, and the reproductive rate. The initial state is always going to be all but one individual is susceptible, therefore all possible stable states were graphed as the depended on only the reproductive rate. Other progress this week included the completion of the first five pages of my report, which involve creating near final versions of my vector fields and numerically integrated graphs. And I also created a viral flow diagram for the SEIRSV2 model. I decided to incorporate High-risk individuals into the model since that was a big feature of COVID-19, causing it to be renamed to SEIRSV2HR. This was done by duplicating the model as seen above. After drawing the viral flow diagram, I realised that I had create six SEIRS loops all interacting with each other. Which will allow for easier organisation and management of differential equations. My goals for next week include implementing the SERISV2HR model in python with the appropriate differential equations, writing further details in my report, and animating the grid based simualtions.

Week 6

This week I achieve my goals in both my software development and mathematical reasoning. In the report, I wanted to include a section that focused on analysing the system of differential equations without solving the system explicitly. I wanted to shows that much of the information that people care about can be obtain from the differential system with the need to solve any definitely equations. In this case, I wanted to show that the system was stable in the endemic phase and unstable with an entirely susceptible population. This was done with Jacobian stability analysis, which involves finding the eigenvalues of a Jacobian matrix evaluated at the points of stability. I was not aware of this technique beforehand; However, I was dimly aware that this kind of stability analysis was possible. I research the method to find out how it worked and applied it to the SIRS model by find the points of stability and the subsequent eigenvalues of the Jacobian matrix. I was fortunate that one of the eigenvalues equalled zero, since that reduced a cubic equation to a quadratic, something I was able to solve easily. When I applied the same method to the SEIRSV2HR model, I unfortunately needed to solve a quartic equation. I decided that I's rather spend my time improving other areas of the report rather than solving the quartic, therefore I resorted to citations and argument to show that the SEIRSV2HR model also approaches a semi-stable state. For my software goals was able to animate the 2D grid simulation and make the boundaries periodic to give the effect of a much larger grid simulation. My goals for next week include writing an analysis of the SEIRSV2HR model in my report, computing an accurate case to death ratio, and implementing the reproductive rate computed by Our World in Data into my models.

Week 7

After making major strides in both my software and report. I realised that my directory structure for my code had become incredible cluttered. What was adequate for a small three compartment simulation became almost impossible to navigate and debug when a system of over 20 compartments was implemented. Initially, the differential equations for the SERISV2HR model were implemented in python in an illogical manner. I found it easier to debug the simulations when I grouped the equations into the 6 loops that were shown in the diagram. Eventually I was able to numerically integrate all the differential equations of the SEIRSV2HR model, and they provided clean and expected results for constant values. This week I also began experimenting with substituting the time dependant reproductive from our world in data into all three compartmental models. While both the SIR and SIRS models provide reasonable accurate results, the SERISV2HR model always gives results that indicate a peak of over a million infections. The results become more in line with the SIR and SIRS results if the latency period is shortened to zero, which implies that there are no programming errors. Instead, the exposed compartment may need to be investigated. However, I'm reluctant to remove it because it was a key feature of COVID-19. Other goals that were completed this week were a calculation of an accurate death to case ration, which I plan to use to estimate the true number of cases during a period where there was limited testing from January to May. My goals for next week include a continuation of the report by writing methods and discussion sections, and investigate the SEIRSV2HR model to obtain more accurate results, and further investigation of the case to death ratio.

Week 8

This week saw a number of course corrections for the project. While I was happy with my results thus far, my supervisor advised me on a supplemental model. I have decided to call this an "Adaptive" model for COVID-19. This is when the coefficients in the model remain constant, including the contact rate. That is until a certain threshold of infections is reach, at which point the contact rate is suddenly reduced. This simulates a government going in and out of lockdown as was seen during the COVID-19 pandemic and is accurate system since lockdowns are put in place in response to infections rather than time. This was implemented easily in my standard numerical integration code with the contact rate being updated in an *if* statement. For the SEIRSV2HR model, the vaccination rate was also updated in this manner with it being set to zero below the threshold and an arbitrary value above zero once passed the threshold, however unlike the contact rate, this was altered permanently. Other goals that were achieved this week was the fact that the grid-based simulations of SIR and SIRS were plotted, and four new compartments were added to the SEIRSV2HR models to account for recoveries of infections for vaccinated individuals. This made the model more realistic, and the symmetry can be leveraged to make the mathematics easier. Further additions to the report were completed, with the first draft of introduction done and the methods section having been begun. My goals for next week include implementing the adaptive model by modifying my current numerical integration code, completing the final graphs and software to be used in the report, and continuing the results section and writing a first draft of the conclusion.

Week 9

This week was a good week for the project because this was the week I completed my software “Experiments”. All my software complied numerical integration that I confirmed was correct and all my grid-based simulations produced results that were in line with the derived differential equations. When creating the numerical integration for the SEIRSV2HR model, I often faced memory and/or overflow errors, therefore my system had to work around these errors as there would be no fix for this bug. When creating the SEIRSV2HR grid-based model the same overflow error occurred, therefore it was decided not to animate this simulation to save resources. There was another bug that prevented individuals from recovering in the grid-based model, however this was fixed after a different implementation was used with the *random.choices()* function used instead of the simple *random()* function to aid in the compactness of the code. Most importantly, I decided on a final name for the SEIRSV2HR model. From now on it will be known as the COVIDGE model, to emphasis the fact that it is a model tailored to COVID-19. My goals for this week include creating a final version of the report, creating a template for the poster, and double checking all my calculations to make sure I haven’t made any mistakes.

Week 10

With the results computed, this week I was able to complete the final sections of the report and poster. I spent about an hour creating a template for the poster, modifying the template given to me by choosing and appealing colour scheme, fonts, and layout. This felt like a nice break as afterwards I completed the results section of my report. Originally, I separated the results up by the models used within them, however, I found the report much easier to write and read when separated by the different versions of the model. Once I had completed the report, I copied various sections of the report into the poster to use as a baseline. I then trimmed, redrafted, and modified the text to fit better within the context of a poster and choose a selection of my results to add to the poster. I then added relevant equations and diagrams, along with some finishing graphical flares. Once a first draft of the poster and report was complete, I read through both to fix as many grammatical errors as possible. I also went through all my mathematical derivations a second time to ensure that I had not made any mistakes. A few key mathematical errors were found in this redraft which would have never been fixed otherwise. Many diagrams were also redrawn to fix errors, which made the report stronger in the end. Both the poster and report were submitted once they had both reached a standard I was content with.