Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project
Final Report

Modeling of in-host viral dynamics using differential equations for estimating uncertainty applied to a classroom

by

Yuanshen Liao

yl5520@ic.ac.uk
GitHub login: acse-yl5520

Supervisors:

Dr. Steven Dargaville
Prof. Christopher Pain

27th August 2021

GitHub repository: github.com/acse-2020/acse2020-acse9-finalreport-acse-yl5520

Abstract

SARS-CoV-2 is a highly infectiousness virus that infects both the upper and the lower respiratory tract. There are lots of studies about the in-host viral dynamic of SARS-CoV-2 infection and how this virus is transmitted and infected across population, but there is a lack of model simulating the its transmission between a small group of people. In this study, we collect different parts from a range of viral dynamic models of SARS-CoV-2, and develop an in-host viral dynamic model which also simulate the concentration of shed virus. We develop Python codes to simulate transmission in a classroom using the model we built by discretizing the classroom into a grid of cells, and uses the forward-backward time stepping method to solve all models as a system of linear equations. We investigate the effectiveness of face mask and vaccine on preventing transmission in a classroom with the simulation we built, and we finds vaccine is better for uninfected students while face mask is better for infected students. Overall, the simulation and model we developed is a framework for simulating viral transmittance across people in a classroom with in-host viral dynamic model as the basis, and our findings prove the health precaution of wearing face mask and getting vaccination is beneficial for people who are required to be in an in-door environment with many others regularly.

1 Introduction

1.1 Motivation

The severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) named by the World Health Organization (2020) is the causative virus of the coronavirus disease (COVID-19) firstly reported in late December 2019 in Wuhan, China (Hui et al. 2020). There are a lot of studies about the in-host viral dynamic of SARS-CoV-2 infection have been done since then, but there is a lack of models simulating the SARS-CoV-2 transmission between a small group of people, e.g. students who are regularly attending lectures in a classroom.

In this study we develop a model which simulates the concentration of virus in a classroom shed by the students, and investigate how the final result is affected by the number of students as well as their health precaution with face mask and vaccination. An objective of this study is to use the developed model to simulate the SARS-CoV-2 transmission between a small group of people, and to quantify the effectiveness of several health precautions which are proposed by governments. Also to provide a scientific evidence from modelling to support the hypothesis that wearing face mask and getting vaccinated has the highest effectiveness.

1.2 Literature Review

A target cell-limited (TCL) model for modelling in-host acute viral infection is introduced by Baccam et al. in 2006. This model is suitable for acute infection where the timescale of infection is significantly shorter than the replenishment of the targeted cells by the virus. The model consists of three ordinary differential equations (ODEs):

$$\begin{aligned}\frac{dT}{dt} &= -\beta VT \\ \frac{dI}{dt} &= \beta VT - \delta I \\ \frac{dV}{dt} &= pI - cV\end{aligned} \tag{1}$$

Target cells T become infected cells I because of the virus V at virus infection rate β. In the process of infection, infected cells die at infected cells elimination rate δ for several reasons: killed by immune system, natural death, etc.; virus is produced by living infected cells at virus production rate p, and die at virus clearance rate c because of similar reasons. The virus clearance rate is assumed as a constant by Baccam et al. for simplicity. This model becomes the basis for many studies which build in-host viral dynamic model for SARS-CoV-2 infection.

Li et al. use the model described in equations 1 to follow the amount of an targeted important cells by SARS-CoV-2 - the pulmonary epithelial cells. These cells are responsible for gas-exchanging in human lung, and are identified as one of the target cells to be infected by SARS-CoV-2 (Wang et al. 2018). They add a constant term to the ODE of target cells to represent the replenishment of pulmonary epithelial cells in lung. This modification increases the sustainability of SARS-CoV-2 within human when the infected cells elimination rate δ is slow, thus credits the elimination of infected cells to the power of the immune system rather than the limited number of target cells (Baccam et al. 2006).

In the pandemic in 2003 China which is caused by the severe acute respiratory syndrome coronavirus (SARS-CoV), some studies detect a significant amount of these virus particle inside a type of white blood cells of SARS-CoV patient (Gu et al. 2005). The genome of SARS-CoV-2 is compared with that of SARS-CoV by some studies in 2020, and they identify there are 82% nucleotide identity between the two virus. Meaning both SARS-CoV and SARS-CoV-2 has a very similar DNA (Chan et al. 2020). Based on this fact Wang et al. build another model based on the same ODEs described in equations 1, estimating the concentration of the pulmonary epithelial cells and the type of white blood cells detected with SARS-CoV - T lymphocytes, in the course of SARS-CoV-2 infection. The model created by Wang et al. is governed by the following four ODEs:

$$\begin{aligned}\frac{dT_1}{dt} &= -\beta V T_1 \\ \frac{dT_2}{dt} &= \lambda - \beta V T_2 \\ \frac{dI}{dt} &= \beta V(T_1 + T_2) - [\delta(t) + \omega T_2]I \\ \frac{dV}{dt} &= pI - cV\end{aligned} \tag{2}$$

Both pulmonary epithelial cells T_1 and T lymphocytes T_2 become infected cells I because of SARS-CoV-2 V at virus infection rate β. During the infection period, T lymphocytes are recruited to the infection site at rate λ, and eliminating infected cells at rate ω. Beside T lymphocytes, infected cells are also eliminated by the rest of the immune system by rate function depends on time $\delta(t)$:

$$\begin{cases}\delta_I, & t < \mu \\ \delta_I e^{\sigma(t-\mu)}, & t \geq \mu\end{cases} \tag{3}$$

δ_I is the basic killing rate of the infected cells. Starts from emergence day μ of the adaptive immune response, this killing rate increases exponentially with a coefficient σ. The ODE for V stays unchanged from the basic model in equations 1.

There are two compartments of human respiratory tract: the upper respiratory tract (URT) and the lower respiratory tract (LRT). This compartment is illustrated in figure 1 taken from Wikipedia. During the model parameters fitting process conducted by Wang et al., they find the parameters in the model have different values when the model is fitted with data collected from the URT through throat swabs and data collected from the LRT through sputum samples. This suggests human body handles SARS-CoV-2 infection differently for the two compartments of respiratory tract. The findings from Wölfel et al. are aligned with this suggestion. They identify target cells of SARS-CoV-2 in both respiratory tracts, and there are some reports showing SARS-CoV-2 causes lung damage in LRT but only mild symptoms in URT.

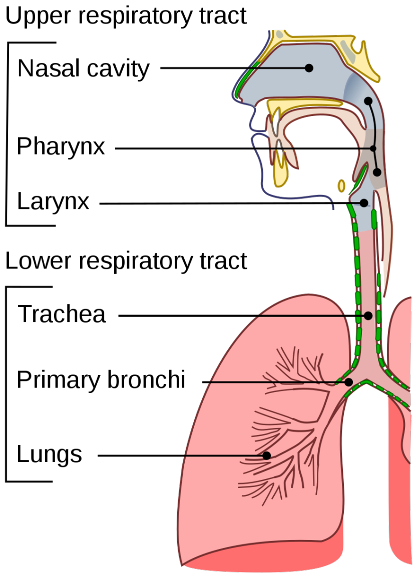

Figure 1: URT and LRT compartments. (Wikipedia 2021)

Ke et al. build another model based on equations 1 to separately simulate early infection of SARS-CoV-2 for both respiratory tracts in one model, and investigate on the relation between in-host viral dynamic and infectiousness. They assume the pulmonary epithelial cells T do not become productive infected cells I after infected with SARS-CoV-2 V, but become eclipse cells E which are infected but not yet productive. A portion of virus from URT is conducted to LRT through the oral-lung aspiration axis (Hou et al. 2020) forming the main source of infection for LRT. Ke et al. fit their model to the same set of data used by (Wang et al. 2020), and find the virus conduction from LRT to URT is negligible. Based on the experimental finding from Leung et al., the amount of virus exhaled by a person is upper bound by a constant even when the person has a large virus load, thus Ke et al. employ a saturated function to estimate the probability of SARS-CoV-2 transmission between people.

2 Software Description

2.1 Overview

We decide to discretize the classroom into a 3-dimensional grid of cells, and assign each of these cells with the model we developed to keep tract on the in-host viral dynamic. Figure 2 shows a $5 \times 3 \times 3$ discretization containing 2 students represented by the two red cells in a simulation.

The model we built simulates the concentration of virus in a classroom shed by the students, and can determine whether a student is effectively infected by others from looking at the simulation results. It is described by the 9 ODEs in equations 4, and figure 3 illustrates the relationship between variables. This model is based on equations 2 but is augmented to simulate both respiratory tracts at the same time like what Ke et al. has done. The differences are:

1. Some students may not be infected at the beginning of our simulation. Therefore, the recruitment of T lymphocytes T_2 in both respiratory tracts becomes a function depends on the corresponding virus concentration V: $\lambda(V(t)) = \begin{cases} 0 & ,V(t) = 0 \\ 10000 & ,V(t) > 0 \end{cases}$.
2. Virus in the environment would infect URT once being inhaled (Leung et al. 2020), this process

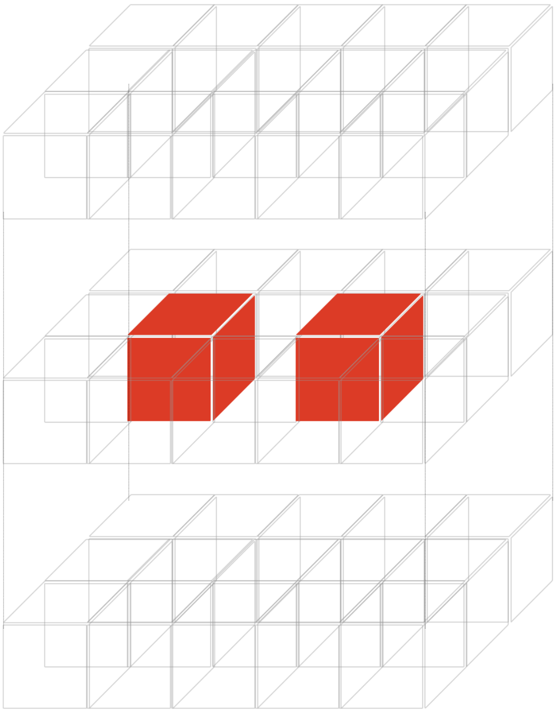

Figure 2: $5 \times 3 \times 3$ discretization with 2 students

is represented by the term $\xi(t)\Gamma_{\text{inhale}}V^E$ in the ODE for V^U. Γ_{inhale} is a multiplier, and function $\xi(t)$ acts like a connector connecting the virus concentration in URT and that in the environment: $\xi(t) = \begin{cases} 0 & \text{, student not in the classroom} \\ 1 & \text{, student in the classroom} \end{cases}$.

3. Virus concentration in URT is conducted to LRT at rate Γ_{conduct}, this is identical to the corresponding term used by Ke et al. in their model.

4. A newly introduced variable V^E which follows the concentration of environmental virus of the cell in the discretized grid.

$$
\begin{aligned}
\frac{dT_1^U}{dt} &= -\beta^U V^U T_1^U \\
\frac{dT_2^U}{dt} &= \lambda^U(V^U(t)) - \beta^U V^U T_2^U \\
\frac{dI^U}{dt} &= \beta^U V^U (T_1^U + T_2^U) - [\delta^U(t) + \omega^U T_2^U] I^U \\
\frac{dV^U}{dt} &= p^U I^U - c^U V^U + \xi(t)\Gamma_{\textsf{inhale}} V^E \\
\frac{dT_1^L}{dt} &= -\beta^L V^L T_1^L \\
\frac{dT_2^L}{dt} &= \lambda^L(V^L(t)) - \beta^L V^L T_2^L \\
\frac{dI^L}{dt} &= \beta^L V^L (T_1^L + T_2^L) - [\delta^L(t) + \omega^L T_2^L] I^L \\
\frac{dV^L}{dt} &= p^L I^L - c^L V^L + \Gamma_{\textsf{conduct}} V^U \\
\frac{dV^E}{dt} &= \xi(t)\Gamma_{\textsf{shed}} V^U - (\sigma_{T\frac{1}{2}} + \sigma_{\textsf{vent}}) V^E + \nabla k^E \nabla V^E
\end{aligned}
\tag{4}
$$

The ODE of the concentration of environmental virus V^E consists three terms:

1. The source term. As implied by the saturated function developed by Ke et al. to estimate the probability of SARS-CoV-2 transmission, the URT virus concentration in the same discretized cell is the only contributor to V^E with $\Gamma_{\textsf{shed}}$ as a multiplier. The same connector function $\xi(t)$ is applied.

2. The elimination term. Environmental virus vanish from the classroom in two ways: ventilation and death, represented by $\sigma_{\textsf{vent}}$ and $\sigma_{T\frac{1}{2}}$ respectively. $\sigma_{\textsf{vent}}$ is proportional to $\frac{\textsf{total flux per day}}{\textsf{volume}}$ of the classroom, and $\sigma_{T\frac{1}{2}}$ can be approximated from the environmental virus half-life, which is about 34 minute for those in a classroom (Dabisch et al. 2021).

3. The diffusion term. Virus in the environment which are not landed on surface will involve in a diffusion process (Dabisch et al. 2021). k^E is a matrix having the same shape as the discretized grid for each variable in our model, each value in this matrix represents the diffusion speed of the corresponding discretized cell of the corresponding variable. These values are set to meet the well-mixed assumption we made in the study. The well-mixed assumption assumes the concentration of environmental virus is evenly spread in the classroom at all time.

We identify the whole piece of work can be modularized into three different units, thus we employ the agile development strategies to build and test each unit of the codes. The validation process of each unit is explained in detail in the following subsections. We develop pieces of Python scripts to conduct the whole simulation with a FORTRAN library, provided by Professor Christopher Pain, which contains subroutine for solving advection-diffusion equations as a system of linear equations as the technical back-end. Figure 4 illustrates an overview structure of the developed codes.

2.2 User Input Preprocessing

We would like to use the developed code to conduct uncertainty quantification for different scenarios, thus we keep the number of hard-coded variables to minimum as expose many important parameters for accepting user input. In this "User Input preprocessing" stage, we load 6 user-configurable `JSON` files to help initiating the simulation:

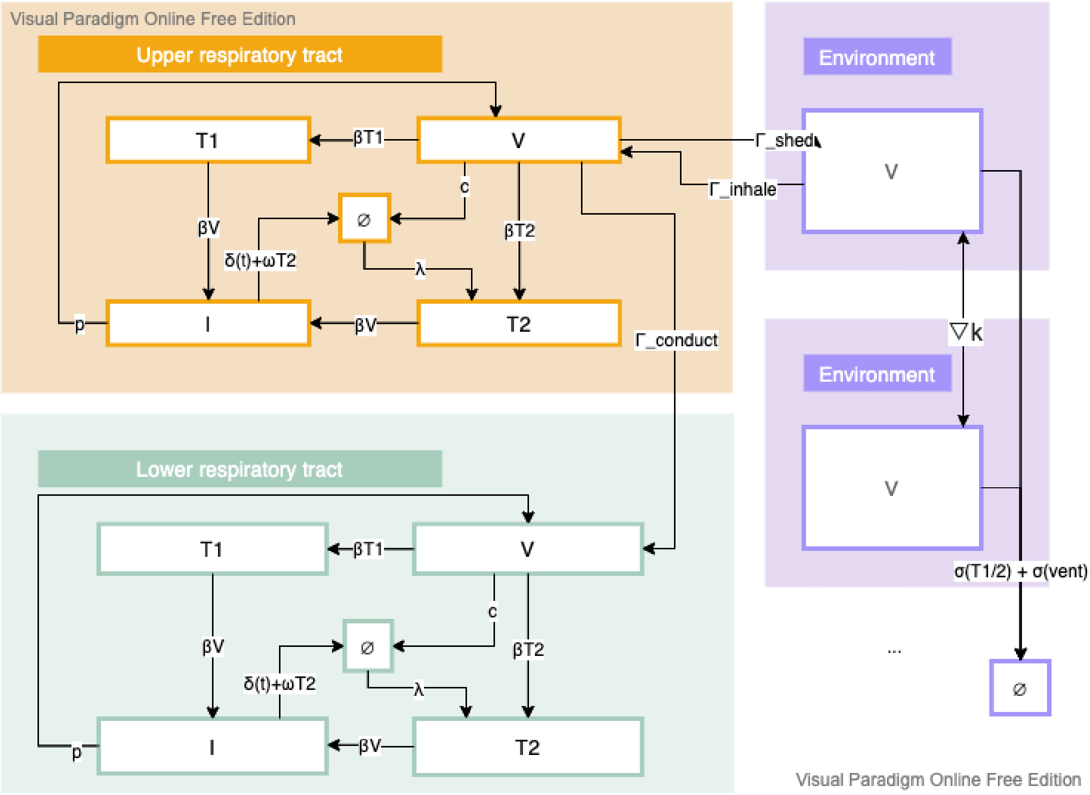

Figure 3: Flow chart of equations 4

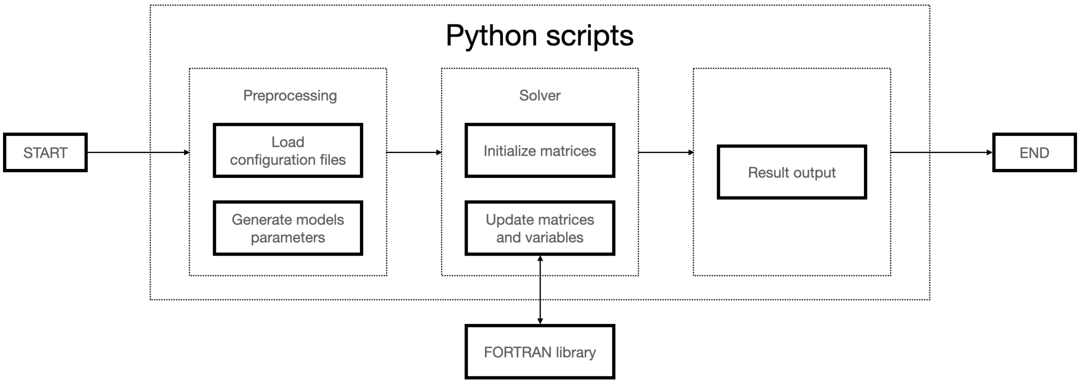

Figure 4: Overview structure.

Type No.	Representation
0	not masked, not vaccinated, not initially infected
1	not masked, not vaccinated, initially infected
2	not masked, vaccinated, not initially infected
3	not masked, vaccinated, initially infected
4	masked, not vaccinated, not initially infected
5	masked, not vaccinated, initially infected
6	masked, vaccinated, not initially infected
7	masked, vaccinated, initially infected

Table 1: Eight attribute groups.

1. constant.json contains the initial values for the in-host viral dynamic variables: T_1^U, T_2^U, etc. The recruitment rate of T lymphocytes λ during infection is also defined in this file.
2. model-conf.json contains the important hyper-parameters for the simulation, e.g. the size of time step in seconds, the duration of the simulation in days.
3. grid-conf.json contains parameters related to V^E, i.e. the initial concentration, the half-life in minutes, and the ventilation ratio.
4. schedule.json defines the periods of time which students are presented in the classroom, the $\xi(t)$ function depends on this definition.
5. distribution.json contains the mean values and standard derivations of model parameters. It is important in generation of these parameters when creating model for different discretized cells.
6. host-conf.json defines the number people in each attribute. We are interested in three aspects for each student: face mask, vaccination, and whether he is infected prior to the beginning of the simulation or not. The combination of the three aspects gives 8 attribute group as described in table 1.

After loading the host-conf.json and distribution.json an instance of the object Hosts is created. The Hosts instance has a variable which holds a list of dictionaries, each of these dictionaries contains its attribute, coordinate in the discretized grid, and all parameters required by the model. These dictionaries are created and appended to the list in the generateHost(genType:int, genAmount:int) python function, this function accepts two integers specifying the attribute type listed in table 1 and the number of such attributes respectively, then calculates or generates values for each key in the dictionary with the corresponding mean values and standard derivations defined in distribution.json.

We decide to simulate all scenarios in a square-shape classroom because it is a more realistic setup, and more importantly there are fewer cells in the discretized grid, compared to a long-shape classroom, when the number of students in the simulation is large. Therefore, the minimum length and width of the discretized grid is calculated by "ceil(sqrt(total_amount))", and the x- and y-coordinate corresponds to the discretized grid of each students can be calculated in Python by "current_list_length // grid_length * 2 + 1" and "current_list_length % grid_length * 2 + 1" respectively.

At this point the size of time step is known from the "model-conf.json", and the scheduling information is loaded from the "schedule.json", it is possible to pre-compute all $\xi(t)$ values for every time step that the simulation will take. Pre-computing these values can avoid keeping track of the actual time information within the iteration process, which eventually gives better performance. The standard module datetime provided in Python helps to compute the $\xi(t)$ values. A list of 0 and 1 which is one-to-one corresponds to each time step is readied, and it is passed to the iteration wrapper function `solver()`.

2.3 FORTRAN library

The subroutine "`sim_time_stepping_diffusion_calc`" in this FORTRAN library, provided by Professor Christopher Pain, is used to calculate the values of all 9 variables in each model assigned to the discretized grid in every time steps. This subroutine first advects and diffuses the values at time step i represented in matrix T_i, then do forward-backward time stepping as linear calculations $T_{i+1} = AT_i + B$ to get the values at next time step. So the validation of this subroutine consists of two parts: the advection-diffusion part and the linear calculation part.

The advection-diffusion part is validated with two criteria

1. This subroutine is designed for closed system, thus the sum of V^E over the whole discretized grid at each time step should not decrease forward in time, if $\sigma_{\text{vent}} = 0$ and $\sigma_{T\frac{1}{2}} = \infty$.
2. With our well-mixed assumption, the value of V^E in each cells should be very near at each time step.

The result acquired from running this subroutine with the parameters and model described by Wang et al. completes the validation of the linear calculation part, by comparing the plot of the result against the plot given by Wang et al..

2.4 Iterative Solver

We have to express the ODEs describing our model as a linear function of the form $T_{i+1} = AT_i + B$ in order to utilize the subroutine in the FORTRAN library provided. As a result:

$$T_i = \begin{bmatrix} T^U_{1_i} & T^U_{2_i} & I^U_i & V^U_i & T^L_{1_i} & T^L_{2_i} & I^L_i & V^L_i & V^E_i \end{bmatrix}^{-1}$$

$$B = \begin{bmatrix} 0 & \lambda^U(V^U_i) & 0 & 0 & 0 & \lambda^L(V^L_i) & 0 & 0 & 0 \end{bmatrix}^{-1}$$

Matrix A is then a sparse matrix with diagonal:

$$-\begin{bmatrix} \beta^U V^U_i & \beta^U V^U_i & \delta^U(t) + \omega^U T^U_{2_i} & c^U & \beta^L V^L_i & \beta^L V^L_i & \delta^L(t) + \omega^L T^L_{2_i} & c^L & \sigma_{T\frac{1}{2}} + \sigma_{\text{vent}} \end{bmatrix}$$

$$A_{20}, A_{21} = \beta^U V^U_i, A_{32} = p^U, A_{38} = \xi(t)\Gamma_{\text{inhale}}$$

$$A_{64}, A_{65} = \beta^L V^L_i, A_{73} = \Gamma_{\text{conduct}}, A_{76} = p^L, A_{83} = \xi(t)\Gamma_{\text{shed}}$$

There are two more importance matrices required by the advection-diffusion part of the subroutine in the FORTRAN library, they are the advection velocity matrix u and diffusion matrix k. For simplicity, we do not consider advection in our model meaning $u = 0$. The well-mixed assumption we forced in the simulation suggests matrix $k \approx$ the length of the diagonal of one layer, and the value on boundary is the negative of the others to enforce a closed system. This magnitude of these values is proven to be suitable by a numerical experiment which is describe in "Implementation & Results" section.

All matrices required by the FORTRAN subroutine are prepared within the Python wrapper function `solver(mConf:Dict, gConf:Dict, hosts:Hosts, xis:List)`, which accepts two dictionaries containing the values loaded from the `JSON` files, an instance of the `Hosts` object, and a list of 0 and 1 values for every time step of $\xi(t)$, this function then starts the iteration process. In each iteration, `solver(...)` does three things in the following order:

1. Updates matrix A and B described above with the current matrix T_i and the current simulation time t.

2. Calls the FORTRAN subroutine with the updated matrices, then replaces the matrix T_i with the matrix T_{i+1} returned by the subroutine.
3. Calls another Python function `extract(...)` to record values in the non-empty cells of T_{i+1} every user-defined number of steps. The empty cells serve no purpose beyond being a separation in the simulation.

Finally, this wrapper function returns all the recorded values and the corresponding simulation times in two arrays.

3 Code metadata

Although Python codes, being an interpreted language, can be ran on all supported platform without compilation (GeeksforGeeks 2021), the provided back-end FORTRAN library has to be recompiled on different processor. The FORTRAN library we used is compiled from a `.f90` source code with the `f2py` module provided in the `Numpy` Python module.

All codes in this study are developed, tested, and run on Mac OS with no specific hardware requirements. However, a more power CPU would speedup the simulation we developed. Up to the current version of code, there is no parallelization technique applied thus number of cores should not affect the performance of the developed code.

The up-to-date version of the developed code is published in

`https://github.com/acse-2020/acse2020-acse9-finalreport-acse-yl5520`

Detail usage guideline is algo provided in the given URL.

4 Implementation & Results

4.1 Model Validation

The validation of our model focuses on the functionality of the coupling terms like Γ_{inhale} which couples V^E to V^U, Γ_{conduct} which couples V^U to V^L, Γ_{shed} which couples V^U to V^E, and the values in k^E which couple V^E in every discretized cell all together. Our decision on focusing the functionality, rather than looking for the most appropriate value, of these coupling terms is based on two reasons:

1. There is a lack of relevant research on quantifying these terms, but the quantification of them is not relevant to our goal.
2. We have no access to enough data for conducting such quantification and, more importantly, we are lack of time.

We check Γ_{inhale}, Γ_{conduct}, and Γ_{shed} works as expected by comparing the simulation results by setting each of them to zero and a positive number. Γ_{inhale} is a source term in the ODE of V^U which accumulates a portion of V^E into V^U. In case of $I^U, V^U = 0$ initially, a correct implementation of this term would results in 0 for both variables during the whole simulation when $\Gamma_{\text{inhale}} = 0$, but this should not be true vice versa, with another student shedding virus in the simulation. For Γ_{conduct}, a similar process is applied. In case of $I^L, V^L = 0$ initially, a correct implementation of Γ_{conduct} would results in 0 for both variables during the whole simulation when $\Gamma_{\text{conduct}} = 0$, but this should not be true vice versa, with I^U, V^U of the same model is not 0 initially. For Γ_{shed}, values of V^E in the whole simulation are the indicators of its validity.

The matrix k^E is responsible to evenly spread the values of V^E across the discretized grid of a closed system, thus its validation consists two aspects:

Name	Representation
Total win	No equilibrium state established in both respiratory tracts
URT win	Only LRT established equilibrium state
LRT win	Only URT established equilibrium state
Lose	Equilibrium state established in both respiratory tracts

Table 2: 4 result cases

1. V^E of each non-boundary cells should be very close to each other at any time step.
2. Without the sink term in V^E ODE, i.e. $\sigma_{T\frac{1}{2}}, \sigma_{\text{vent}}$, the sum of V^E from all discretized cells should not decrease throughout the simulation.

We test k^E with different values for different size of discretized grid, and we observe the two aspects are met when the magnitude of k^E values is close to the double of the length of the discretized grid diagonal. For example for the given grid shown in figure 2, each layer has 5×3 cells and each cells is 0.1×0.1 (hard-coded values), this gives the diagonal a length of $\sqrt{(5 \times 0.1)^2 + (3 \times 0.1)^2}$. Thus for this given grid, a k^E with magnitude ≈ 1 passes our validation.

4.2 Code Performance

The most computational part in the simulation is the "Solver" module, as depicted in figure 4, this module iteratively does two things: update values in matrices and update matrix by doing matrix arithmetic. The matrix arithmetic is done in the FORTRAN library provided and there are some optimizations applied during compilation by the f2py module from Numpy. We use numpy.ndarray to store the matrices, this data type is proven to be more efficient than the native List type provided in Python.

4.3 Results

The simulation outputs a .pkl file produced by the pickle module from Python, values of each variable in our model at every output time step are recorded in this file for all non-empty discretized cells. We say an equilibrium state is reached between T_2 and the concentration of SARS-CoV-2 V in URT or LRT of a discretized cell if:

1. There is no more pulmonary epithelial cell T_1.
2. The concentration of SARS-CoV-2 is just enough to infect all the recruited T lymphocytes.

Visually, a plot of such cell looks like the plot at the bottom in figure 5. The orange line represents the concentration of URT T lymphocytes, and the line does not go up or down by the end of simulation on day 30. As comparison, the plot on top is a discretized cell without an equilibrium state being developed. The concentration of URT T lymphocytes is still rising.

Based on the establishment of the equilibrium state, we define and name 4 cases to classify the results in simulation as shown in table 2. Among all 4 cases, "Total win" case is valued the most since this indicates the corresponding student is not infected at all or is recovered in both respiratory tracts by the end of our simulation; "URT win" case has more value than "LRT win", since it indicates the student cannot shed a significant amount of virus to the environment, and thus less infectiousness; "Lose" case comes at last.

Finally, we construct experiments to investigate how effective face mask and vaccination is by comparing the amount of each cases from two aspects: 1. level of protection on 63 initially uninfected students (type 0, 2, 4, and 6) from 1 type 1 student (type 1); 2. level of protection on 63 type 6 students from 1 initially infected students (type 1, 3, 5, and 7). Table 1 describes all types that we are referring here.

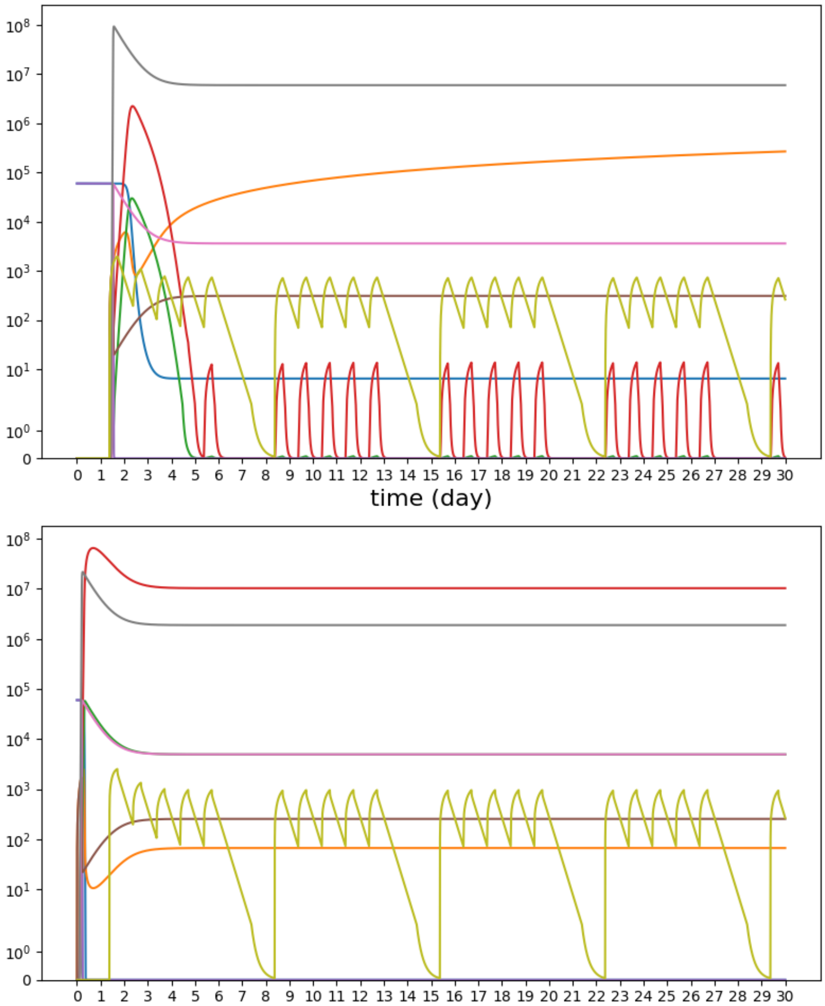

Figure 5: Example plots from the `.pkl` file

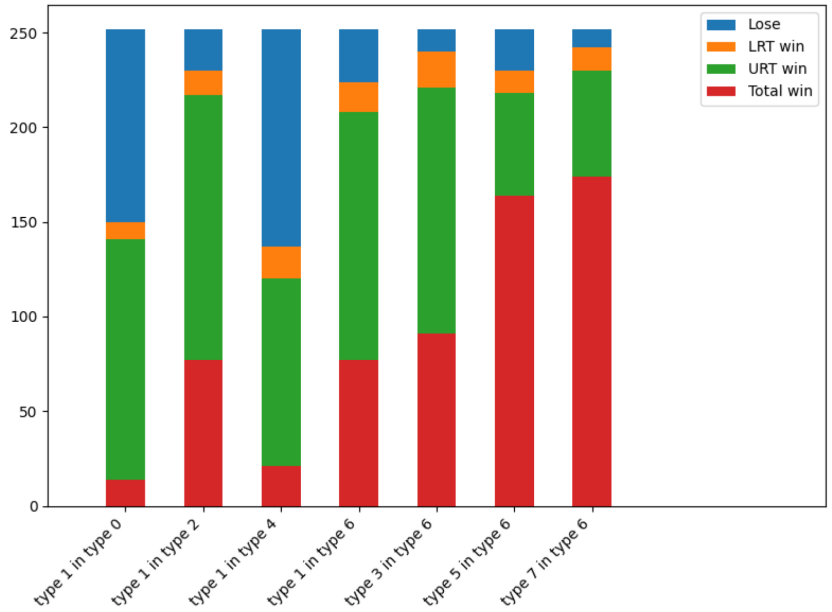

Figure 6: Number of each type of cases in experiments

In each experiment, there is only one type of initially infected students and only one type of initially uninfected students. We run 4 separate simulations for each combination of student types and sum the number cases.

The four bars on the left in figure 6 shows the numbers of cases count in each type, for a type 1 student being simulated with 63 initially uninfected students in a classroom. Based on the these 4 bars, student types which have the most number of "Total win" cases are type 2 and 6, these students are all vaccinated in common. More students who are without vaccination end up in "Total win" cases if they are wearing a face mask (type 4). These results reflect and compare the importance of health precautions:

- Without face mask and vaccination, students are more unlikely to prevent infection or recover from infection, when staying in the same classroom with an infection students.
- Although face mask provides protection to some extend, vaccination gives students much more protection in the same situation.

The four bars on the right in figure 6 shows the numbers of cases count in each type, for 1 initially infected student being simulated with 63 type 6 students in a classroom. The increasing number of "Total win" cases indicate all uninfected students are more unlikely to be infected, as the level of health precaution taken by the infected student increases. Different from the previous comparison between health precaution taken by uninfected students, this result suggests face mask is a more effective precaution for infected students on protecting uninfected students.

5 Discussion & Conclusions

In summary, we develop an in-host viral dynamic model to calculate the concentration of pulmonary epithelial cell as T_1, T lymphocytes as T_2, infected cells as I ans SARS-CoV-2 as V in both compartment of respiratory tracts, as well as the concentration of shed virus as V^E. Then we develop Python codes to use a provided FORTRAN library to simulate transmission in a classroom using the in-host viral dynamic model we built. The simulation first discretizes a classroom into a grid of cells and integrate a model we developed to each of these cells, with some cells being non-empty to represent a student by initializing the variables and selecting values from random distributions for the parameters of the integrated model. Finally, uses the forward-backward time stepping method in the provided FORTRAN library to iteratively solve all models concurrently as a system of linear equations. We use the developed simulation to investigate the effectiveness of face mask and vaccine on preventing transmission in a classroom, we conclude from the simulation results that vaccine is more effective than face mask for uninfected students to protect themselves from being infected, while face mask is a better option for infected students to protect others from being infected by them.

When we are investigating the simulation result, we find the concentration of virus V increases much quicker in the very beginning of infection on those initially uninfected students than the initially infected students. This difference is induced by the different concentration of environmental virus at different time point. We set the initial concentration of environmental virus to 0 for all simulation, meaning initially infected students do not inhale virus but only shed. By the time the initially uninfected students are getting infected, the concentration of environmental virus is already high in value, thus the concentration of virus in the URT of these students accumulate faster than earlier in time. This suggests people who are infected in a contaminated in-door environment would a shorter incubation period (duration between infection and symptom onset).

The main problem and limitation of our simulation is the random distributions which we select model parameters from. The mean values and standard derivations are calculated from the data provided by Wang et al., these data are fitted from COVID-19 patients which is bias to reflect the real value for a more general population. For contrasting the distribution between different student attributes/types, we shift the mean values of some distributions according to some researches which are not directly related to these parameters, e.g. a research report comparing number of the infected people with and without vaccine (Imperial College London 2020), and a study about amount of virus detected in URT of people with and without face mask (Leung et al. 2020). We believe a more accurate distribution can increase the accuracy of our simulation.

References

Baccam, P., Beauchemin, C., Macken, C. A., Hayden, F. G. & Perelson, A. S. (2006), 'Kinetics of influenza a virus infection in humans', *Journal of virology* **80**(15), 7590–7599.

Chan, J. F.-W., Kok, K.-H., Zhu, Z., Chu, H., To, K. K.-W., Yuan, S. & Yuen, K.-Y. (2020), 'Genomic characterization of the 2019 novel human-pathogenic coronavirus isolated from a patient with atypical pneumonia after visiting wuhan', *Emerging Microbes & Infections* **9**(1), 221–236.

Dabisch, P., Schuit, M., Herzog, A., Beck, K., Wood, S., Krause, M., Miller, D., Weaver, W., Freeburger, D., Hooper, I. et al. (2021), 'The influence of temperature, humidity, and simulated sunlight on the infectivity of sars-cov-2 in aerosols', *Aerosol Science and Technology* **55**(2), 142–153.

GeeksforGeeks (2021), 'Difference between compiled and interpreted language', `https://www.geeksforgeeks.org/difference-between-compiled-and-interpreted-language/`. Accessed: 2021-08-15.

Gu, J., Gong, E., Zhang, B., Zheng, J., Gao, Z., Zhong, Y., Zou, W., Zhan, J., Wang, S., Xie, Z., Zhuang, H., Wu, B., Zhong, H., Shao, H., Fang, W., Gao, D., Pei, F., Li, X., He, Z., Xu, D., Shi, X., Anderson, V. M. & Leong, A. S.-Y. (2005), 'Multiple organ infection and the pathogenesis of sars', *Journal of Experimental Medicine* **202**(3), 415–424.

Hou, Y. J., Okuda, K., Edwards, C. E., Martinez, D. R., Asakura, T., 3rd, K. H. D., Kato, T., Lee, R. E., Yount, B. L., Mascenik, T. M., Chen, G., Olivier, K. N., Ghio, A., Tse, L. V., Leist, S. R., Gralinski, L. E., Schäfer, A., Dang, H., Gilmore, R., Nakano, S., Sun, L., Fulcher, M. L., Livraghi-Butrico, A., Nicely, N. I., Cameron, M., Cameron, C., Kelvin, D. J., de Silva, A., Margolis, D. M., Markmann, A., Bartelt, L., Zumwalt, R., Martinez, F. J., Salvatore, S. P., Borczuk, A., Tata, P. R., Sontake, V., Kimple, A., Jaspers, I., O'Neal, W. K., Randell, S. H., Boucher, R. C. & Baric, R. S. (2020), 'Sars-cov-2 reverse genetics reveals a variable infection gradient in the respiratory tract', *Cell* **182**(2), 429–446.

Hui, D. S., Azhar, E. I., Madani, T. A., Ntoumi, F., Kock, R., Dar, O., Ippolito, G., Mchugh, T. D., Memish, Z. A., Drosten, C., Zumla, A. & Petersen, E. (2020), 'The continuing 2019-ncov epidemic threat of novel coronaviruses to global health — the latest 2019 novel coronavirus outbreak in wuhan, china', *International Journal of Infectious Diseases* **91**, 264–266.

Imperial College London (2020), 'Coronavirus infections three times lower in double vaccinated people - react', `https://www.imperial.ac.uk/news/227713/coronavirus-infections-three-times-lower-double/`. Accessed: 2021-08-15.

Ke, R., Zitzmann, C., Ribeiro, R. M. & Perelson, A. S. (2020), 'Kinetics of sars-cov-2 infection in the human upper and lower respiratory tracts and their relationship with infectiousness', *medRxiv* .

Leung, N. H., Chu, D. K., Shiu, E. Y., Chan, K.-H., McDevitt, J. J., Hau, B. J., Yen, H.-L., Li, Y., Ip, D. K., Peiris, J. M. et al. (2020), 'Respiratory virus shedding in exhaled breath and efficacy of face masks', *Nature medicine* **26**(5), 676–680.

Li, C., Xu, J., Liu, J. & Zhou, Y. (2020), 'The within-host viral kinetics of sars-cov-2', *Mathematical Biosciences and Engineering* **17**(4), 2853–2861.

Wang, S., Pan, Y., Wang, Q., Miao, H., Brown, A. N. & Rong, L. (2020), 'Modeling the viral dynamics of sars-cov-2 infection', *Mathematical Biosciences* **328**, 108438.

Wang, Y., Tang, Z., Huang, H., Li, J., Wang, Z., Yu, Y., Zhang, C., Li, J., Dai, H., Wang, F., Cai, T. & Tang, N. (2018), 'Pulmonary alveolar type i cell population consists of two distinct subtypes that differ in cell fate', *Proceedings of the National Academy of Sciences* **115**(10), 2407–2412.

Wikipedia (2021), 'Respiratory tract', https://en.wikipedia.org/wiki/Respiratory_tract. Accessed: 2021-08-15.

World Health Organization (2020), 'Naming the coronavirus disease (covid-19) and the virus that causes it', https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it. Accessed: 2021-08-15.

Wölfel, R., Corman, V. M., Guggemos, W., Seilmaier, M., Zange, S., Müller, M. A., Niemeyer, D., Jones, T. C., Vollmar, P., Rothe, C., Hoelscher, M., Bleicker, T., Brünink, S., Schneider, J., Ehmann, R., Zwirglmaier, K., Drosten, C. & Wendtner, C. (2020), 'Virological assessment of hospitalized patients with covid-2019', *Nature* **581**, 465–469.