Journal of Computational Science 61 (2022) 101662

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

# GPU acceleration and data fitting: Agent-based models of viral infections can now be parameterized in hours

Baylor G. Fain, Hana M. Dobrovolny *

*Department of Physics and Astronomy, Texas Christian University, Fort Worth, TX, United States of America*

ARTICLE INFO

*Keywords:*
Agent-based model
Viral infection
Graphical processing unit
Computer simulation

ABSTRACT

For many years, infectious disease modelers have used agent-based models to study the spread of viruses, but the models were too computationally intensive to fully replicate even in vitro experiments. Now, with technological advancements and accessible software, agent-based models can be used to their full potential. In this work, we created an agent-based model that expresses viral transmission and diffusion, can manipulate and track individual cells, and can be fit to real experimental data in a timely manner due to acceleration of computation with graphics processing units (GPUs). The use of GPUs allows simulations to run on desktop computers in a few minutes or hours, while still simulating an accurate number of cells to replicate infection experiments. This model can now be used to study in-host infections quickly, so that in the event of an outbreak or epidemic a treatment plan and course of action can be developed in less time.

## 1. Introduction

Agent-based (individual-based or micro-simulation) models have been around since 1970 with the introduction of "Conway's Game of Life" [1]. These models have been utilized in many different fields from physics to the study of fish (ichthyology) [2] and continue to be popularized for different applications by software like Netlogo [3,4]. The models consist of a collection of agents whose behavior is determined by mathematical or computational rules. The agents of the system can move freely [5] or be fixed in a grid or lattice [6] for varying applications, but either configuration allows for tracking of spatially emergent patterns. In recent years, the field of virology has started using agent-based models to study the spread of viruses in a monolayer of cells [6–13] in an effort to study the spatiality of viral spread.

In a lab, *in vitro* viral infections are performed on layers of cells grown to the point of confluence, where there is on the order of $10^5$–$10^6$ cells [14]. Virus modelers are using ABMs to simulate the two dimensional layer of agents to replicate experiments that are done *in vitro* in order to better understand factors that affect viral spread. The agent-based model framework is appealing to virus modelers because it allows for the individual tracking of how cells, as agents, interact with the virus, and has the potential to replicate *in vitro* and eventually *in vivo* viral infections. Currently, however, the implementation of agent-based models in the field of virology has two issues: speed and size.

Agent-based models are notorious for being computationally intensive and taking long amounts of time to run simulations. This point has been commented on in a previous article [15] and the feasibility of ABMs for research has been talked about as a goal that is to come with increasing computational advancements [16]. Previous research has addressed this lack of computing power issue by reducing the number of agents modeled and therefore reducing the number of computations required for a simulation. In other in-host studies, the number of agents published is at minimum an order of magnitude lower than the number of target cells used in the corresponding experimental data. Beauchemin et al. [6] simulated $1.232 \times 10^5$ agents, while the experiment they were attempting to replicate was performed in 6 well-plates and had $\sim 1.2 \times 10^6$ cells per well. Alvarado et al. [7] simulated $4.0 \times 10^4$ agents when trying to replicate experiments also performed in 6 well-plates. Wodarz et al. [8] simulated $2.0 \times 10^4$ agents, while the experiment they were replicating was performed in 24 well-plates and had $\sim 2.4 \times 10^5$ cells per well. Tong et al. [9] simulated $6.0 \times 10^5$ agents in an effort to simulate mice lungs, which have $\sim 10^9$ cells. These smaller simulations are more affected by boundary interactions, which can result in model dynamics that do not faithfully reproduce the infection. This can hinder the research process and can make it difficult to use models for treatment optimization and personalized medicine.

While it might be feasible to wait long periods of time to run accurate simulations for endemic or recurrent seasonal viruses, recent events of the COVID-19 pandemic indicate how great a need there is for accurate and fast modeling methods. Epidemiological population-level modeling tools that include both ordinary differential equation

* Corresponding author.
*E-mail address:* h.dobrovolny@tcu.edu (H.M. Dobrovolny).

https://doi.org/10.1016/j.jocs.2022.101662
Received 28 June 2021; Received in revised form 29 March 2022; Accepted 31 March 2022
Available online 9 April 2022

(ODE) models [17,18] and ABMs [19–21] were immediately deployed to help predict how the new virus would spread around the world and how different interventions could help stem the spread. At the within-host level, the primary modeling tool was limited to simple ODE models [22–25] that lack the ability to reproduce the spatial heterogeneity of real viral infections. Of the ABMs, Getz et al. [26] is a community-driven ABM, incorporating many realistic biological responses, that was quickly developed for SARS-CoV-2, but is only currently simulating a few thousand agents and is expected to need high-performance computing or cloud resources to parameterize the model. Fast and accurate in-host models could be helpful in assessing the potential of re-purposed drugs [22,27,28], finding indicators of disease severity or mortality [29], and assessing the effectiveness of testing [30]. As the field of virology continues to move towards methods of modeling like ABMs for in-host viral dynamics, researchers need to be aware of how to increase computational performance in accessible ways and potentially with equipment that is already in the lab.

In this paper, we present the testing and validation of a combination agent based model (ABM) and partial differential equation model (PDEM) implemented on GPUs, which allows for rapid simulation of large-scale ABMs on desktop computers. The work here begins with the methods, where the four attributes of the model — the agent-based model of the cells, the partial differential equation model of the virus, the cell-free transmission mode of viruses, and fitting of the model to data — are explained. We then present the results of model implementation with parallel programming, convergence testing, and simulation speed improvement. Finally, we show that the model can reproduce experiments by fitting the model to an example data set from an *in vitro* influenza experiment. Allowing for this work to be an example, the main contributions to computational science and the main takeaways for researchers are to call attention to the current and increasing power of desktop GPUs, and that ABMs of large and complex systems can be simulated in reasonable amounts of time using desktop GPUs without the need of a "super computer".

## 2. Methods

### 2.1. Model details

In this work, a two dimensional biological system is simulated with a mathematical model. The system is a culture dish of a monolayer of cells with virus diffusing over the cells. The model is a combination of an agent based model (ABM) and a partial differential equation model (PDEM) where the cells are represented with an ABM and virus diffusion is represented by a PDEM.

#### 2.1.1. Viral transmission

When a virus is spreading among the cells in a culture dish, there is a probability that a healthy cell becomes infected by virus that is not within a cell, but flowing around and above the cell. When this viral transmission occurs it is called cell-free transmission. For cell-free transmission, the probability ($P_{cf}$) that a cell becomes infected is determined by the amount of virus that is covering the cell ($V$), the infection rate ($\beta$) [31], and the time step ($\Delta t$),

$$P_{cf} = V\beta\Delta t.$$

As a healthy cell becomes surrounded by more virus, the probability of cell-free infection increases. If, due to the build up of virus, $P_{cf}$ ever becomes greater than one, the time step is divided in half until the probability is back below one. Then for each moment the time step was halved, the new probability of cell-free infection is compared with a number from the uniform distribution. If during any of the comparisons the number is less than the new probability of cell-free infection, then the cell becomes infected.

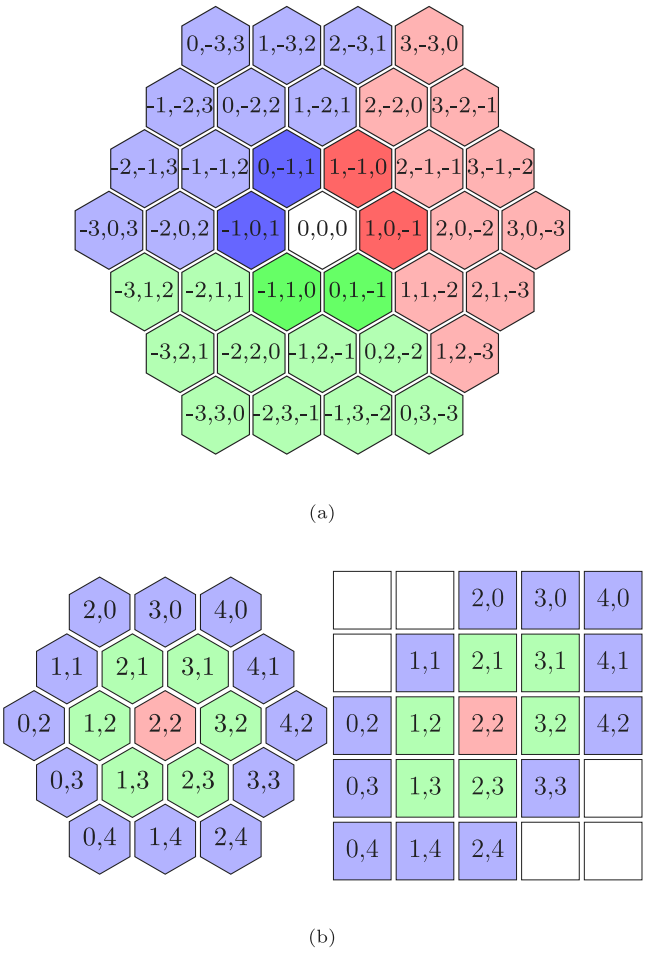


**Fig. 1.** Here the attributes of hexagonal coordinates are illustrated. The attributes are used to save time by either reducing the number of calculations needed or the amount of searching through data arrays. Figure (a) shows that the three different regions are cyclic permutations of each other, therefore reducing the number of agent locations needed to be calculated. Figure (a) also shows, in the darker shaded regions, that the neighbors of a hexagon can be found by adding a cyclic permutation of the two vectors $\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$. Figure (b) shows that the locations of the hexagons can be stored in a square matrix that can easily be uploaded into computer memory.

#### 2.1.2. Spatial accounting

To allow for the two dimensional aspect of the culture dish to be represented in the model, the cells are approximated as hexagons. Using hexagons allows for an elegant managing of the cells' shapes in the dish and the viral transmission. Since the culture dishes are grown to confluence, the cells are close enough that they push on each other and the cell walls deform. This causes the cells to no longer be in the shape of a circle, but become irregular polygons with multiple sides [32]. Modeling the cells as hexagons gives the cells definite sides and the cells are able to span any two dimensional region forming a hexagonal grid. Furthermore, by using a hexagonal grid, when virus particles spread among this population of cells the indexing of the grid can be used to find the neighbors of any cell. This will be used for cell-free transmission to know where virus will flow away from (high concentrations areas) and to (low concentration areas) during diffusion.

In addition to helping with the physical representation of the model, hexagonal coordinates (often represented by vectors $\begin{bmatrix} q \\ r \\ s \end{bmatrix}$, with $q$, $r$, and $s$ directions) have some other attributes that can be utilized to

optimize the code for quicker compute times. The three attributes that this code utilizes are:

1. The coordinates can be split in to three sectors where the hexagonal coordinates $q$, $r$, and $s$ are simply cyclic permutations.
2. The neighbors of a particular hexagon can be found by adding, to that hexagon's coordinates, a cyclic permutation of $\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ for three of the neighbors and $\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$ for the other three neighbors.
3. The $q$ and $r$ hexagonal directions can be used as indices of a matrix.

These attributes save time by either reducing the number of calculations needed or the amount of searching through data arrays. Specifically, Attribute 1 reduces the number of cell locations that need to be calculated by a third, as seen in Fig. 1(a), where the coordinates of the hexagons in each of the three sections (red, green, or blue) are a cyclic permutation of the coordinates of the hexagons in a different section. Attribute 2 gives the coordinates of the neighboring hexagons to a particular hexagon by adding to that hexagon's coordinates a cyclic permutation of $\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ or $\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$. This can be seen in the darker hexagons of Fig. 1(a) for the hexagon $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$. Attribute 3 allows the hexagon locations to be easily stored and referenced in the memory of a computer. Fig. 1(b) illustrates Attribute 3 by showing the locations of the hexagons mapped to a square grid using the $q$ and $r$ coordinate of hexagonal coordinates.

#### 2.1.3. Agent-based model

In an ABM, a system is broken down into smaller units called "agents". Each of the agents are governed by a set of rules on a local scale with large scale phenomena resulting from interaction of the agents, so the two scales are studied to find the connections. As a simulation of the model is stepped through time, the agents act and interact. These actions cause bulk properties, that may appear disconnected from the individual agents, to manifest. Properties are observed and measured to find the connection between the small interactions and large scale properties.

In this work, an ABM governs the transitions a cell makes through the stages of infection: healthy, eclipse, infected, and dead. A cell in the healthy state is an uninfected cell that remains healthy until infected. A cell in the eclipse state is an infected cell that is not yet producing virus. The cell remains in the eclipse state for an average amount of time, $\tau_E$. The specific time value for each cell is determined by a gamma distribution with shape value $\eta_E$ and scale value $\tau_E/\eta_E$. A cell in the infected state is an infected cell that is producing virus. The cell remains in the infected state for an average amount of time, $\tau_I$. The specific time value for each cell is determined by a gamma distribution with shape value $\eta_I$ and scale value $\tau_I/\eta_I$. A gamma (Erlang) distribution is used for the amount of time in the eclipse and infected phase, as suggested by the work of Beauchemin et al. [33] and Kakizoe et al. [34]. A cell in the dead state is a cell that can no longer change state, so once a cell is in the dead state the cell remains in that state until the end of the simulation. The flow of this is illustrated in Fig. 2.

The ABM uses four time arrays to track and transition the cells to different states after infection. The four arrays universal time (UT), healthy time (HT), eclipse time (ET), and infected time (IT) have an element for each cell. The universal time array holds the amount of time that each cell has been in the simulation; each element starts at zero and increases each iteration by the simulation's time step. The healthy time array holds the amount of time that a cell is healthy; each element starts at zero and while the cell is healthy increases each iteration by the simulation's time step. The eclipse time array holds the amount of time each cell is in the eclipse state and the infected time array holds the amount of time each cell is in the infected state. For the eclipse and infected arrays the amount of time is fixed and the value is determined by a gamma (Erlang) distribution, as described above. The flow of this is illustrated in Fig. 2.

#### 2.1.4. Partial differential equation model

PDEMs are used to model multiple dimensions; in this work a PDE in hexagonal coordinates is used to model the two-dimensional spatial spread of virus over cells in a culture dish. In a PDEM, the dynamics of a system can be represented by a partial differential equation, or more specifically, an equation that contains multi-variable functions that represent important system aspects and one or more partial derivatives of those functions. In the culture dish, as an infected cell releases virus into the extracellular fluid, the virus diffuses across a concentration gradient. The PDEM represents this movement with the diffusion equation,

$$\frac{\partial V}{\partial t} = D\nabla^2 V + p - cV, \tag{1}$$

where $V$ is the density of the virus, $D$ is the diffusion coefficient, $p$ is the production rate per cell, and $c$ is the viral clearance rate. In the code, along with the assumption of hexagonal cells, the cells are assumed to be flat, so the virus is diffusing over a smooth two dimensional plane. This assumption allows for the use of the two dimensional diffusion equation in hexagonal coordinates, so Eq. (1) becomes

$$\frac{\partial V}{\partial t} = D\frac{2}{3}\left(\frac{\partial^2}{\partial q^2} + \frac{\partial^2}{\partial r^2} + \frac{\partial^2}{\partial s^2}\right)V + p - cV$$

where $q$, $r$, and $s$ are the hexagonal directions and $\hat{\mathbf{q}} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$, $\hat{\mathbf{r}} = \frac{\sqrt{2}}{2}\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$, and $\hat{\mathbf{s}} = \frac{\sqrt{2}}{2}\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$ are the unit vectors for a hexagonal grid in hexagonal coordinates. For computation we solve the diffusion equation, with Neumann boundary conditions, in hexagonal coordinates using a forward Euler numerical scheme on the same grid with the same grid spacing as the ABM.

#### 2.1.5. Parameters of viral spread

The eight parameters $\beta$, $\tau_E$, $\eta_E$, $\tau_I$, $\eta_I$, $p$, $c$, and $D$ affect the dynamics of virus spread in the model. Four of the parameters, $\tau_E$, $\eta_E$, $\tau_I$, and $\eta_I$, are used in the ABM to choose the time duration that a cell is in the eclipse and infected phase as mentioned in Section 2.1.3. Three of the other parameters, $p$, $c$, and $D$, are used in the PDEM and characterize the differential equation, as mentioned in Section 2.1.4. The final parameter, $\beta$, governs the interaction between the virus and cells, setting the probability that the cell is infected. In order to model a particular virus, values for these parameters need to be chosen. The initial values of the parameters are chosen from ordinary differential equation models of influenza and listed in Table 1 (viral titer units have been converted to virions, as described in [35]).

### 2.2. Computational details

#### 2.2.1. Implementation on GPUs

As the model becomes more complex, GPU acceleration via parallel programming is used to decrease the simulation run times and therefore increase the number of studies that can be conducted in a given time. In the simulations, the cells change state based on the amount of virus above them. The number of cells in a culture dish is on the order of $10^6$ cells [14], so the ABM will simulate a grid of 1001365 agents of hexagonal cells in a circle to best replicate what is happening in the experiment. Each agent will follow the rules of checking the amount of

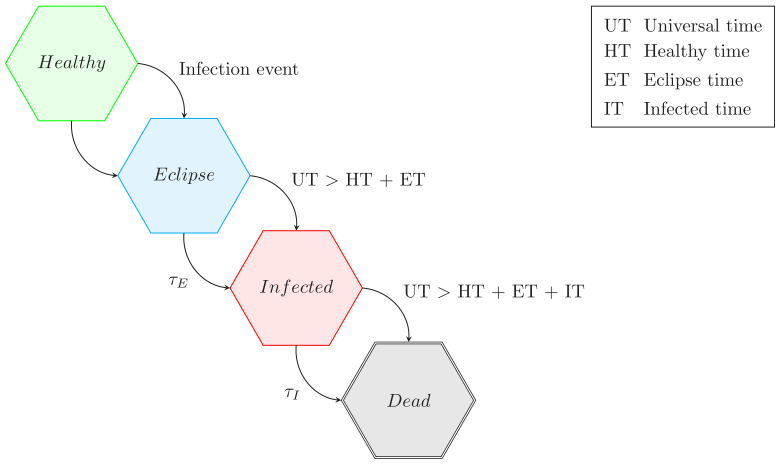


**Fig. 2.** The stages of infection: healthy, eclipse, infected, and dead are shown. The cells transition through the stages at different time points. Above: The time point when a state transition occurs is shown in terms of UT, the universal time, for a cell. Below: The time point when a state transition occurs is shown in terms of average time. $\tau_E$ is the average time a cell stays in the eclipse stage and $\tau_I$ is the average time a cell stays in the infected stage.

**Table 1**
Parameter values to simulate an influenza infection with the ABM/PDEM combo model.

| Parameter | Meaning | Value | Reference |
|---|---|---|---|
| $\beta$ | Infection rate | 2.0/h | Scaled from Beauchemin et al. [36] |
| $p$ | Viral production rate | 562800/h | Scaled from Beauchemin et al. [36] |
| $c$ | Viral clearance rate | 0.105/h | Beauchemin et al. [36] |
| $D$ | Diffusion coefficient | $2.16\times10^{-8}$ m$^2$/h | Stokes–Einstein equation |
| $\tau_E$ | Mean eclipse duration | 6.0 h | Beauchemin et al. [36] |
| $\eta_E$ | Eclipse shape parameter | 30 | Pinilla et al. [37] |
| $\tau_I$ | Mean infectious lifespan | 12.0 h | Beauchemin et al. [36] |
| $\eta_I$ | Infectious shape parameter | 100 | Pinilla et al. [37] |

virus above the cell every time step. Utilizing attribute 3 of hexagonal coordinates, the number of calculations is reduced from the order of ($\mathcal{O}(n^2)$) per time step to the order of the number of agents ($\mathcal{O}(n)$). The calculations from the agents' rules are split over the processing units of a GPU to be calculated in parallel or simultaneously. To utilize parallel processing, Nvidia's CUDA (Compute Unified Device Architecture) is used to implement the ABM and PDEM. CUDA is an Application Programming Interface (API) that allows the many processing units (cores) on a Nvidia brand GPU to be used for computing.

To determine the gain in computation speed by use of GPUs, codes utilizing serial and parallel processing are compared. For the serial processing, two codes were written (not utilizing a threading framework, API, or package) one in Python and one in C. Python and C are two commonly used programming languages in physics. For parallel processing, a C code was written utilizing CUDA. The amount of computation time needed to simulate one hour of the infection was measured and compared.

### 2.2.2. Convergence testing

Partial differential equations (PDEs) are a popular way to model systems that evolve over both space and time, but often require computers to produce solutions. With PDEs, even systems that have an exact solution often need to be calculated on a computer because of the infinite series that are required in those solutions. Therefore, solutions to PDEs are often found through numerical integration. In the numerical integration, space and time are discretized, that is, they are assumed to be made up of small units. From this discretization, time is a one dimensional line of points separated by a chunk of time called $\Delta t$ and two dimensional Cartesian space is a grid with a line of points for each dimension where there is a chunk of space for each dimension $\Delta x$, $\Delta y$. At these points in time and space, a numerical integration scheme approximates the solution of the PDE. Different numerical schemes have different benefits. Depending on the phenomena that needs to be studied with the PDE the size of $\Delta t$, $\Delta x$, and $\Delta y$ and the choice of numerical scheme are important. If the chunks of space or time are too large then the simulation does not have the resolution to resolve phenomena that occur at smaller increments in the model and if the numerical scheme requires too much computing power then the solutions cannot be found in a timely manner.

Depending on the choice of numerical scheme, a conditional relationship between $\Delta t$, $\Delta x$, and $\Delta y$ must be met. For the symmetric, two dimensional Euler's method

$$\Delta t \leq \frac{(\Delta x)^2}{4D},$$

is the conditional relationship [38,39]. Satisfying this relationship is necessary to ensure that the sequence of approximations that the numerical scheme uses to approximate a solution converges, otherwise the error grows exponentially to a point that the solutions are unreliable. Using the relationship above, values for $\Delta t$, $\Delta x$, and $\Delta y$ can be chosen to ensure stability of the error in the numerical scheme. As long as that relationship is met, the solution is reliable within a certain error, but the relationship does not give the $\Delta t$, $\Delta x$, and $\Delta y$ that are best for producing accurate simulations with the least amount of computing cost.

To ensure the simulation of the PDEM converges, we need to optimize the space and time discretizations: $\Delta t$, $\Delta x$, and $\Delta y$. Convergence testing is a simple brute force method where the input parameters are increased or decreased by a particular amount and the accuracy or trends of the simulation are measured for each of the new increments. Schemes for convergence testing are implemented and studied in fields like computational fluid dynamics [40,41] and astrophysics [42,43]. The model proposed in this work has fixed $\Delta x$ and $\Delta y$ to a value of 50 μm, because the simulations are of real cells, whose average diameter can be measured between 50–100 μm. Thus the convergence testing only has to be conducted for $\Delta t$. To conduct the study a starting point of 0.005 h, about 5.78 times smaller than the conditional relationship of $\frac{25}{864} \approx 0.0289$ h, was chosen and a range of seven values was created by

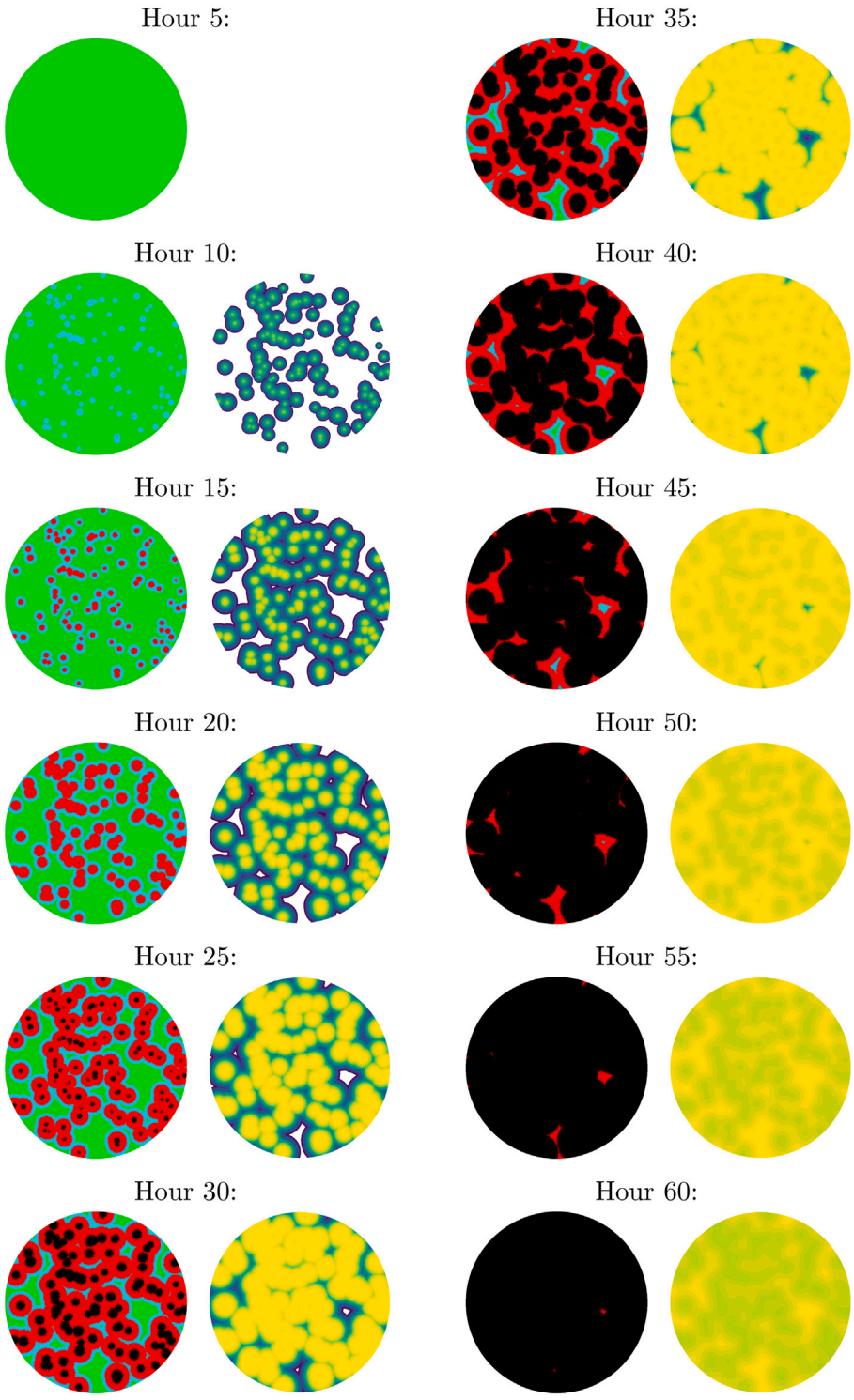


**Fig. 3.** The dish at hours 5 through 60 in 5 h increments. On the left are cells in the different stages of infection; the stages are represented by healthy cells colored green, eclipse cells colored cyan, infected cells colored red, and dead cells colored black. On the right are images of the virions that are diffusing over the cells; areas of higher concentration are represented by yellow and areas of lower concentration are represented by purple.

multiplying or dividing the initial $\Delta t$ by 2 repeatedly. For each of these $\Delta t$s, the median viral titer curves of ten simulations were compared.

### 2.3. Data fitting

As part of our model validation, we verified that the model could reproduce viral titer curves observed experimentally. The experimental data set used here is from an *in vitro* experiment performed by Pinilla et al. [37]. During the study, a well of a 24-well plate, containing Madin–Darby canine kidney (MDCK$\alpha$2,6) cells was inoculated with the A/Québec/144147/09 (H1N1) pandemic strain of influenza virus and the supernatant fluid was collected every 6 h until 36 h and then every 12 h until 72 h post infection. The supernatant was then used for RNA isolation and/or viral titration by standard plaque assay on MDCK$\alpha$2,6 cells. The specific data referenced for this work is the multiple-cycle viral yield experiment shown in figure 2A of the Pinilla manuscript.

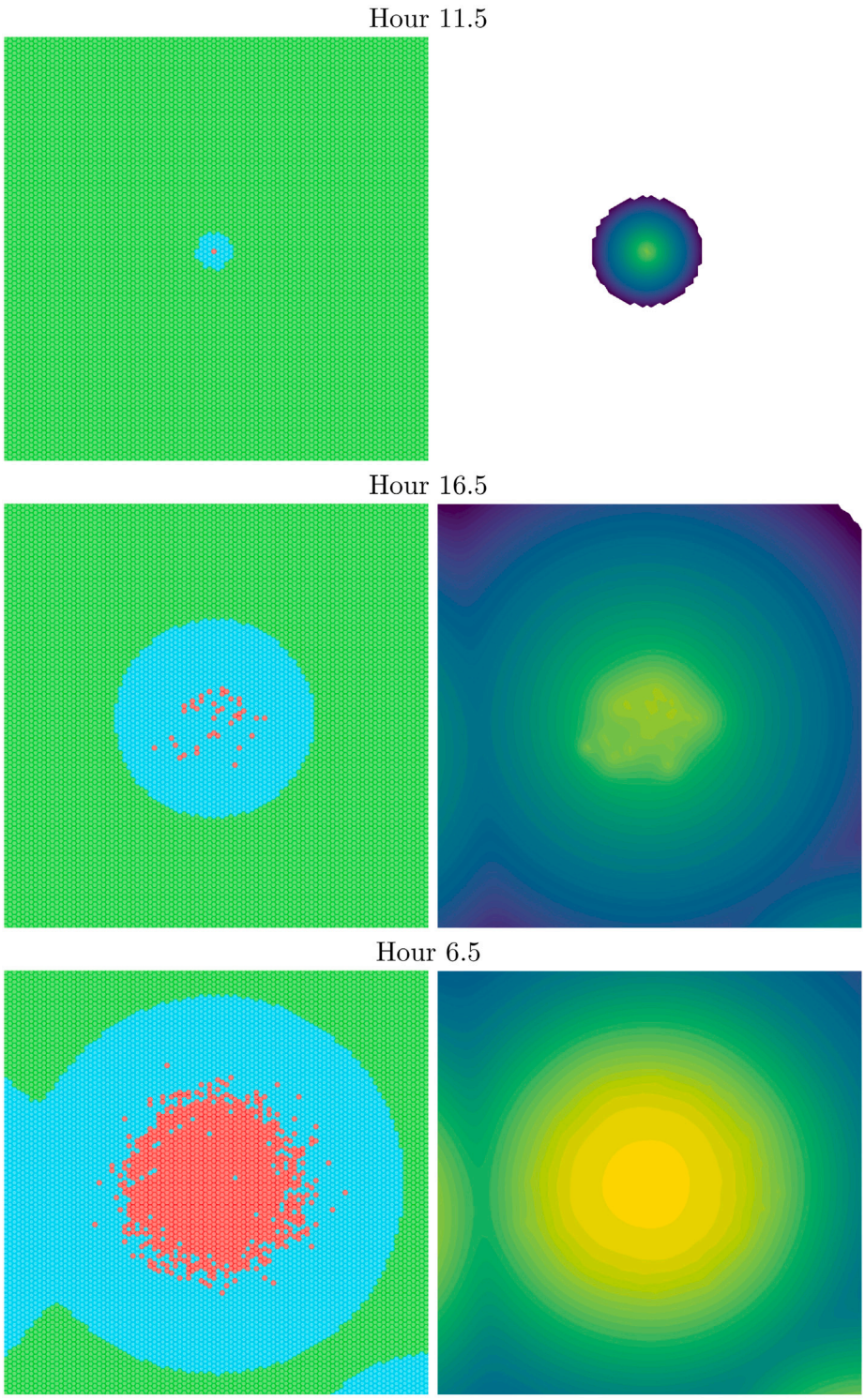


**Fig. 4.** A zoomed in section of the dish looking at the plaque formed by a single infected cell during a viral infection at hours 6.5, 11.5, and 16.5. On the left are cells in the different stages of infection; the stages are represented by healthy cells colored green, eclipse cells colored cyan, infected cells colored red, and dead cells colored black. On the right are the many virus that are diffusing over the cells; areas of higher concentration are represented by yellow and areas of lower concentration are represented by purple.

To determine the best fit of the model to the experimental data, the sum of square residuals (SSR) is minimized,

$$\mathrm{SSR} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

where $y_i$ is from the experimental data set and $\hat{y}_i$ is from the simulated data set. In our case, the simulated data set is the average of ten cell-free transmission simulations. The initial conditions, of the cells and virus, for each of the ten simulations are: Total cells — 1001365, Total virus — 0.0, and Number of cells in the Eclipse phase — 10. To perform the minimization, a separate code that utilizes the function `minimize` from the python package `scipy`, was written. In the code, five parameters ($\beta$, $p$, $\tau_I$, $\tau_E$, and $c$) are allowed to vary and the remaining parameters are held fixed to the values given in Table 1. The minimization code is given an initial guess for the five parameters, then

by the Nelder–Mead method the next set of parameters is produced, until the minimum SSR is found.

## 3. Results

### 3.1. Model simulation

Using the influenza parameters of Table 1, we simulated an infection initiated with 1001365 cells in a dish, 100 cells in the eclipse phase, and no initial virus. Fig. 3 shows different views of plaques forming in the entire dish. On the left are cells in the different stages of infection described in Section 2.1.3, where the healthy cells are colored green, eclipse cells are colored cyan, infected cells are colored red, and dead cells are colored black. On the right are figures showing the corresponding virus concentrations that are over the cells, where areas of higher concentration are colored yellow and areas of lower concentration are colored purple. Fig. 3 shows the infection in 5 h increments starting at 5 h, when no cells are producing virus, and ending at 60 h, when almost all the cells have died. The ABM reproduces the plaques typically seen in experimental *in vitro* infections [44].

For a closer look at the plaques, Fig. 4 is a zoomed in view of the infection at hours 6.5, 11.5, and 16.5. The cells are shown on the left, with the same color scheme used in Fig. 3, and the corresponding virus distribution is shown on the right. Here we can clearly see the heterogeneous growth of the plaque; it is not simply a radially symmetric change of cells from eclipse to infectious.

### 3.2. Implementation on GPUs

To show how implementing GPUs could potentially increase the computation speed for a researcher, we simulated ten viral infections for five different numbers of cells, with codes that utilize three different programming languages: Python, C, and CUDA. These codes were run on a single desktop computer built with an Intel Xeon E-2144G CPU, 16 GB of RAM, and a P4000 Nvidia Graphics card. The amount of computation time needed to simulate one hour of the infection is shown in Fig. 5. The compute times for the three codes increase as the number of cells in the simulations increases. The speed increase of switching from Python, the programming language commonly used in physics, to code written in C is about 40 times faster, similar to results found in a general study of implementation of ABMs on GPUs [45], and the speed increase is about 7000 times faster when switching to code using CUDA for implementation on GPUs. These results show that researchers can achieve a substantial computational speed increase by simply implementing simulations of models on GPUs either directly with CUDA (as done in this work) or using an API or software like OpenACC, OpenMPI, and FLAME GPU.

#### 3.2.1. Convergence testing

We examine three scenarios when testing the convergence of the model: an infection initiated with 10013 cells in the eclipse phase (Initial Cell); an infection initiated with $10^{12}$ virions (Initial Virus); and a scenario with no infection, but $10^{12}$ virions (Only Virus), examining viral spread and decay only. Simulations in each of the scenarios used the influenza parameters from Table 1. Fig. 6 shows the simulations of the three scenarios, where the time step was varied to test the convergence of the model in time. A time step of 0.005 h, about 5.78 times smaller than the conditional relationship from Section 2.2.2, was chosen and a range around it was made by dividing or multiplying by 2 repeatedly. This formed an array of seven time step values, 0.000625, 0.00128, 0.0025, 0.005, 0.01, 0.02, and 0.04 h, where all of the time steps except 0.04 h are below the conditional relationship. For each time step, the median curve of ten viral titer curves is shown in Fig. 6, from left to right: curves of a viral infection initiated with infected cells; curves of a viral infection initiated with virus; and curves of virus without underlying cell infection. We see that for all the time steps, except 0.04 h, the curves are hard to distinguish from one another and follow the same trend for each scenario. These tests show that choosing a time step value of 0.005 h for simulations is a selection that is well below the conditional relationship, but also reproduces results that are similar to simulations that were run with a time step slightly greater than or slightly lower than 0.005 h.

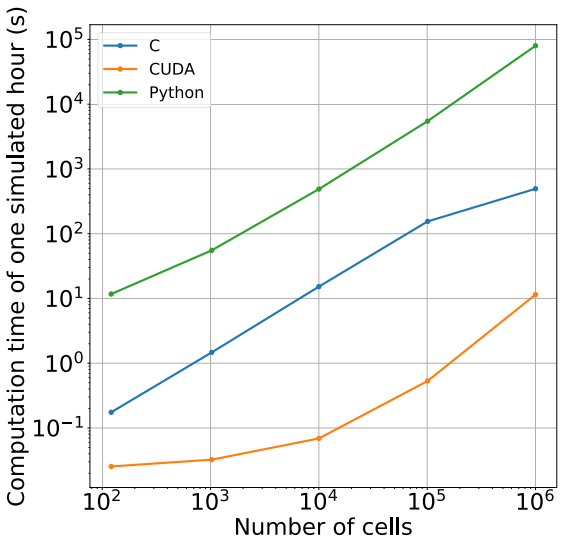


**Fig. 5.** With more than a million cells, CUDA is about 7000 times faster then the Python code and about 40 times faster than the C code.

**Table 2**
Best fit parameter values from fitting model to experimental data [37].

| Parameter | Meaning | Value |
|---|---|---|
| $\beta$ | Infection rate | 54/h |
| $p$ | Viral production rate | 3000/h |
| $c$ | Viral clearance rate | 0.25/h |
| $D$ | Diffusion coefficient | $2.2\times10^{-8}$ m$^2$/h (fixed) |
| $\tau_E$ | Mean eclipse duration | 16 h |
| $\eta_E$ | Eclipse shape parameter | 30 (fixed) |
| $\tau_I$ | Mean infectious lifespan | 26 h |
| $\eta_I$ | Infectious shape parameter | 100 (fixed) |

### 3.3. Fitting the model to data

We fit the model to experimental *in vitro* data [37] via minimization of the SSR. The initial conditions of the simulations were: 501535 cells in the dish (similar to the number of cells in a typical 24-well plate [14]), 50 cells in the eclipse phase, and no virus in the dish. In Fig. 7, the median curve of ten simulations, using the best fit parameters, is plotted in dark blue alongside the data in green. Our best fit parameters are presented in Table 2. Although the same data was used, some of our parameter estimates differ from those reported in Pinilla et al. [37]. Our best fit $\tau_I$ is smaller than the $\tau_I = 49\,\text{h}$ reported by Pinilla et al. while our best fit $\tau_E$ is larger than the $\tau_E = 6.6\,\text{h}$ found by Pinilla et al. and the best fit $c$ is larger than $c = 0.13\,\text{h}^{-1}$. Some of this discrepancy might be due to the inclusion of spatial effects in the ABM, but Pinilla et al. also used more data — they used both a single cycle and multiple cycle experiment as well as viral RNA measurements — to constrain the parameter estimates. All in all, the ABM/PDEM combo model can replicate the viral titer measurements of a typical infection via fitting where the fitting process uses standard packaged fitting algorithms and the computational time for fitting is less than 24 h from initial guess to best fit.

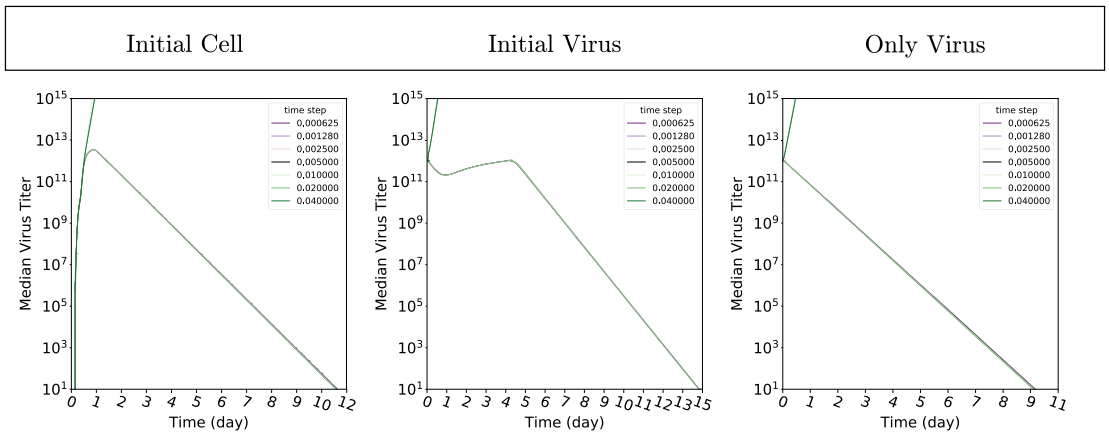


**Fig. 6.** The time step was varied to test the convergence of the model in time. A time step of 0.005 h was chosen and a range around it was made by dividing or multiplying by 2 repeatedly. Seven values were used [0.000625, 0.00128, 0.0025, 0.005, 0.01, 0.02, 0.04]. The median curve of ten viral titer curves is shown for each time step. From left to right, curves of a viral infection exhibiting cell-free transmission initiated with infected cells; curves of a viral infection exhibiting cell-free transmission initiated with virus; and curves of virus without underlying cell infection.

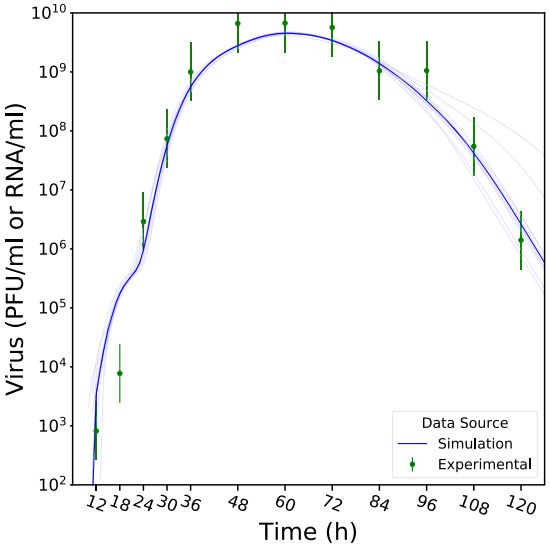


**Fig. 7.** The ten simulated titer curves and corresponding median curve are plotted in blue. The experimental cell-free transmission data [37] is plotted in green. The median curve has the minimal SSR with respect to the experimental data, when using the best fit parameters. The best fit parameters are shown in Table 2.

## 4. Discussion

In this paper, we describe the construction of an ABM/PDEM combo model that is accelerated by GPUs to investigate spatially extended in-host viral infections. Before this work, in virology and epidemiology, GPUs have only been used to accelerate population level ABM models [46–48]. While other previous ABM/PDEM combo models of viral spread [6,16] have explored in-host viral spread, these works only utilized serial processing frameworks and therefore lack the computing power to parameterize their models with accurate numbers of cells. The formulation of the model here, as described in the methods, implements GPUs and vastly improving the simulation speed of these models. This allows for efficient replication of *in vitro* infections with a realistic number of cells. This will help lead to a better understanding of virus-cell dynamics *in vitro* [49], but could also help realize the goal of simulating infections *in vivo* [50]. The faster simulations also allowed us to use standard ODE model-fitting techniques to fit this model to experimental data, making it possible to quickly parameterize these models to reproduce dynamics of different viruses. This work should encourage researchers to either directly implement GPUs in their modeling efforts or utilize APIs and software like OpenACC, OpenMPI, and FLAME GPU to decrease their computational time. Previously, researchers have had to develop other techniques to help speed up fitting of ABMs to experimental data, including reducing the sampled parameter space [51], and mapping of ABM outputs to simpler functions [9,52]. These techniques coupled with simulation of ABMs on GPUs could potentially allow for real-time parameter estimation of models for use in patient care. This is particularly useful for a novel pandemic virus that can be simulated such that trial runs of test drugs can be performed and viral infection severity for a patient can potentially be predicted.

Although our model currently only incorporates cell-free transmission, since the ABM models interactions of each cell in a culture dish, the spatial aspects of different viral transmission routes can be explored in detail. There has been recent interest in viruses that transmit via cell to cell transmission, with ODE [53–55], stochastic [56], and ABM [49,57] models developed to study how cell to cell transmission alters infection dynamics. There are also viruses that cause cells that form syncytia, which are cells that have fused into a single multi-nucleated cell. Not much is known about how syncytia alter infection dynamics, with a recent ODE model attempting to assess the effect of syncytia on viral time course [58], but spatial effects really need to be included for a proper assessment of the role of syncytia. Finally, advection can be added to the diffusion of the virus particles to more closely mimic the respiratory tract. Recent PDE [59] and ODE [60] models both indicate that the addition of advection can limit the spread of respiratory viruses towards the lower respiratory tract, but the stochasticity included in an ABM might affect this result. Each of the model extensions mentioned above can be easily added to the existing model by either adding, respectively: a new `if` statement, a new array for specific information for each cell, or a new term to the diffusion equation.

While the model is able to replicate a typical viral time course during an infection, it is missing many components that play important roles in the infection. For example, the immune response of the host has not been added to the model. The immune response is a large, if not the main, contributing factor to symptoms experienced during a viral infection [61,62], but also limits spread of infection itself [63]. ABMs are already used to model various aspects of the immune response [10,64,65], so the immune response can be incorporated into the existing ABM/PDEM combo framework. Cell tropism, the preference of virus for one cell type over another, is another feature of viral infections that can be incorporated into the ABM. ODE modeling indicates that cell tropism can lead to longer lasting infections [66], but will also likely affect the spatial dynamics of infection. Finally, variation in production of virus

**Table 3**
A list of GPU specifications, assuming a memory usage of 100 MB per million cells.

| GPU | Memory (MB) | # of cells per card |
|---|---|---|
| Quadro p400 | 2 | $2.0 \times 10^7$ |
| Quadro p4000 | 8 | $8.0 \times 10^7$ |
| Quadro rtx 8000 | 48 | $4.8 \times 10^8$ |

by individual cells [67] can be incorporated to determine how this type of cell heterogeneity affects spatiotemporal infection dynamics.

Though attributes like syncytia formation, immune response, and an even more detailed cell tropism representation are still to be added to the current model, there is an ample amount of memory on a modern standard GPU. The amount of memory that the proposed model uses for $10^6$ cells is 34 MB. This means a GPU with 8 GB of RAM running the model could support $235 \times 10^6$ cells or ~235 individual simulations of *in vitro* experiments. In adding the above attributes to the model, the model would use an additional 20 MB, by likely over-estimating 8 MB for syncytia, 8 MB for an immune response, and 4 MB for further cell tropism details. This results in an estimated memory usage of 54 MB, so for the same GPU with 8 GB of RAM mentioned above, the GPU could support $148 \times 10^6$ cells or ~148 individual simulations of *in vitro* experiments.

With the ever increasing computational needs of society and medical care, hospitals will have many computers with a GPU already built in. If we assume a future model might need 100 MB of memory per one million cells, Table 3 shows that even a single Quadro p400 has enough memory to simulate $20 \times 10^6$ cells or ~20 individual 35 mm simulations of *in vitro* experiments. This means that even one of the smallest GPUs that Nvidia recommends (as of 2021) would have more than enough space for future growth. Continuing to think forward, a human respiratory model would need to simulate $4.0 \times 10^8$ cells [68] and therefore take up 40 GB of memory on a GPU. Once *in vivo* simulations are implemented, the Quadro rtx 8000 (a three year old GPU as of 2021) has enough memory to simulate the human respiratory model. Therefore GPUs already have enough memory to store and run viral models of this type. So as GPUs become more commonplace in computers and as they become more powerful, hospital and doctor office computers will naturally have enough computing power to simulate individual viral courses for patient care.

## 5. Conclusion

In this paper, we describe the use of GPUs to accelerate computation of an agent-based and partial-differential equation model. This allows for simulation results within hours, but with the necessary level of detail to capture individual cell effects, and allows for parameterizing the model quickly. The model in this work accurately replicates the diffusion of a virus, the stages of infection of individual cells, and can be fit to data within hours. While still lacking some of the biology needed for replication of *in vivo* infections, the speed of computation leaves room for incorporation of additional features. Because of the acceleration gain from GPUs, this model implementation forms the foundation of a modeling and simulation tool that can accurately predict in-host viral dynamics and be quickly deployed to help combat the next pandemic. Furthermore, this work shows as an example to future researchers that great computation power is available, via desktop GPUs, without the need of a "super computer".

## CRediT authorship contribution statement

**Baylor G. Fain:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft preparation, Writing – review & editing. **Hana M. Dobrovolny:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability statement

ABM/PDEM combo, simulation code and Data-Fitting code are available at https://github.com/BaylorFain/Fain-Dobrovolony-2021-ABM-GPU-DataFitting.git.

## Acknowledgment

All authors approved the version of the manuscript to be published.

## References

[1] M. Gardner, Fantastic combinations of john conway's new solitaire game of life, Sci. Am. 223 (4) (1970) 120, http://dx.doi.org/10.1038/scientificamerican1070-120.
[2] K.A. Owusu, E. Acevedo-Trejos, M.M. Fall, A. Merico, Effects of cooperation and different characteristics of marine protected areas in a simulated small-scale fishery, Ecol. Complex. 44 (2020) 100876, http://dx.doi.org/10.1016/j.ecocom.2020.100876.
[3] D.D. Nogare, A.B. Chitnis, Netlogo agent-based models as tools for understanding the self-organization of cell fate, morphogenesis and collective migration of the zebrafish posterior lateral line primordium, Seminars Cell Dev. Biol. 100 (2020) 186–198, http://dx.doi.org/10.1016/j.semcdb.2019.12.015.
[4] F. Chiacchio, M. Pennisi, G. Russo, S. Motta, F. Pappalardo, Agent-based modeling of the immune system: Netlogo, a promising framework, Biomed. Res. Intl. 2014 (2014) 907171, http://dx.doi.org/10.1155/2014/907171.
[5] C. Beauchemin, N. Dixit, A.S. Perelson, Characterizing t cell movement within lymph nodes in the absence of antigen, J. Immunol. 178 (9) (2007) 5505–5512, URL http://www.jimmunol.org/cgi/content/abstract/178/9/5505.
[6] C. Beauchemin, J. Samuel, J. Tuszynski, A simple cellular automaton model for influenza A viral infections, J. Theoret. Biol. 232 (2) (2005) 223–234, http://dx.doi.org/10.1016/j.jtbi.2004.08.001, URL https://linkinghub.elsevier.com/retrieve/pii/S0022519304003819.
[7] A. Alvarado, R. Corrales, M.J. Leal, A. De la Ossa, R. Mora, M. Arroyo, A. Gomez, A. Calderon, J.L. Arias-Arias, Cellular-level characterization of Dengue and Zika virus infection using multiagent simulation, in: 2018 IEEE International Work Conference on Bioinspired Intelligence, IWOBI, IEEE, San Carlos, 2018, pp. 1–6, http://dx.doi.org/10.1109/IWOBI.2018.8464219, URL https://ieeexplore.ieee.org/document/8464219.
[8] D. Wodarz, C.N. Chan, B. Trinite, N.L. Komarova, D.N. Levy, On the laws of virus spread through cell populations, J. Virology 88 (22) (2014) 13240–13248, http://dx.doi.org/10.1128/JVI.02096-14, URL https://jvi.asm.org/content/88/22/13240.
[9] X. Tong, J. Chen, H. Miao, T. Li, L. Zhang, Development of an agent-based model (ABM) to simulate the immune system and integration of a regression method to estimate the key ABM parameters by fitting the experimental data, PLoS One 10 (11) (2015) e0141295, http://dx.doi.org/10.1371/journal.pone.0141295, URL https://dx.plos.org/10.1371/journal.pone.0141295.
[10] J. Whitman, A. Dhanji, F. Hayot, S.C. Sealfon, C. Jayaprakash, Spatio-temporal dynamics of host-virus competition: A model study of influenza A, J. Theoret. Biol. 484 (2020) 110026, http://dx.doi.org/10.1016/j.jtbi.2019.110026.
[11] A. Goyal, J.M. Murray, Modelling the impact of cell-to-cell transmission in hepatitis B virus, PLoS One 11 (8) (2016) e0161978, http://dx.doi.org/10.1371/journal.pone.0161978.
[12] J. Itakura, M. Kurosaki, Y. Itakura, S. Maekawa, Y. Asahina, N. Izumi, N. Enomoto, Reproducibility and usability of chronic virus infection model using agent-based simulation; comparing with a mathematical model, Biosystems 99 (1) (2010) 70–78, http://dx.doi.org/10.1016/j.biosystems.2009.09.001.
[13] S. Wasik, P. Jackowiak, M. Figlerowicz, J. Blazewicz, Multi-agent model of hepatitis C virus infection, Arti. Intel. Med. 60 (2) (2014) 123–131, http://dx.doi.org/10.1016/j.artmed.2013.11.001.
[14] Thermofisher, Useful Numbers for Cell Culture - US, https://www.thermofisher.com/us/en/home/references/gibco-cell-culture-basics/cell-culture-protocols/cell-culture-useful-numbers.html, library Catalog: www.thermofisher.com.
[15] M. Gallagher, C. Brooke, R. Ke, K. Koelle, Causes and consequences of spatial within-host viral spread, Viruses 10 (11) (2018) 627, http://dx.doi.org/10.3390/v10110627, URL http://www.mdpi.com/1999-4915/10/11/627.
[16] A.L. Bauer, C.A. Beauchemin, A.S. Perelson, Agent-based modeling of host–pathogen systems: The successes and challenges, Inform. Sci. 179 (10) (2009) 1379–1389, http://dx.doi.org/10.1016/j.ins.2008.11.012, URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2731970.

[17] R. Li, S. Pei, B. Chen, Y. Song, T. Zhang, W. Yang, J. Shaman, Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV2), Science http://dx.doi.org/10.1126/science.abb3221.

[18] C.N. Ngonghala, E. Iboi, S. Eikenberry, M. Scotch, C.R. MacIntyre, M.H. Bonds, A.B. Gumel, Mathematical assessment of the impact of non-pharmaceutical interventions on curtailing the 2019 novel coronavirus, Math. Biosci. 325 (2020) 108364, http://dx.doi.org/10.1016/j.mbs.2020.108364.

[19] F. Ying, N. O'Clery, Modelling COVID-19 transmission in supermarkets using an agent-based model, PLoS One 16 (4) (2021) e0249821, http://dx.doi.org/10.1371/journal.pone.0249821.

[20] K. Sneppen, B.F. Nielsen, R.J. Taylor, L. Simonsen, Overdispersion in COVID-19 increases the effectiveness of limiting nonrepetitive contacts for transmission control, Proc. Natl. Acad. Sci. USA 118 (14) (2021) e2016623118, http://dx.doi.org/10.1073/pnas.2016623118.

[21] T. Kano, K. Yasui, T. Mikami, M. Asally, A. Ishiguro, An agent-based model of the interrelation between the COVID-19 outbreak and economic activities, Proc. R. Soc. A 477 (2245) (2021) 20200604, http://dx.doi.org/10.1098/rspa.2020.0604.

[22] A. Gonçalves, J. Bertrand, R. Ke, E. Comets, X. de Lamballerie, D. Malvy, A. Pizzorno, O. Terrier, M.R. Calatrava, F. Mentré, P. Smith, A.S. Perelson, J. Guedj, Timing of antiviral treatment initiation is critical to reduce SARS-CoV-2 viral load, CPT Pharmacomet. Syst. Pharmacol. 9 (9) (2020) 509–514, http://dx.doi.org/10.1002/psp4.12543.

[23] S. Wang, Y. Pan, Q. Wang, H. Miao, A.N. Brown, L. Rong, Modeling the viral dynamics of SARS-CoV-2 infection, Math. Biosci. 328 (2020) 108438, http://dx.doi.org/10.1016/j.mbs.2020.108438.

[24] E.A. Hernandez-Vargas, J.X. Velasco-Hernandez, In-host modelling of COVID-19 kinetics in humans, medRxiv, http://dx.doi.org/10.1101/2020.03.26.20044487.

[25] P. Dogra, J. Ruiz-Ramírez, K. Sinha, J.D. Butner, M.J. Pelaez, M. Rawat, V.K. Yellepeddi, R. Pasqualini, W. Arap, H.D. Sostman, V. Cristini, Z. Wang, Innate immunity plays a key role in controlling viral load in COVID-19: mechanistic insights from a whole-body infection dynamics model, ACS Pharmacol. 3 (1) (2020) 248–265, http://dx.doi.org/10.1021/acsptsci.0c00183.

[26] M. Getz, Y. Wang, G. An, M. Asthana, A. Becker, C. Cockrell, N. Collier, M. Craig, C.L. Davis, J.R. Faeder, A.N.F. Versypt, T. Mapder, J.F. Gianlupi, J.A. Glazier, S. Hamis, R. Heiland, T. Hillen, D. Hou, M.A. Islam, A.L. Jenner, F. Kurtoglu, C.I. Larkin, B. Liu, F. Macfarlane, P. Maygrundter, P.A. Morel, A. Narayanan, J. Ozik, E. Pienaar, P. Rangamani, A.S. Saglam, J.E. Shoemaker, A.M. Smith, J.J. Weaver, P. Macklin, Iterative community-driven development of a SARS-CoV-2 tissue simulator, BioRXiv, http://dx.doi.org/10.1101/2020.04.02.019075.

[27] P. Czuppon, F. Debarre, A. Goncalves, O. Tenaillon, A.S. Perelson, J. Guedj, F. Blanquart, Success of prophylactic antiviral therapy for SARS-CoV-2: Predicted critical efficacies and impact of different drug-specific mechanisms of action, PLoS Comput. Biol. 17 (3) (2021) 1–19, http://dx.doi.org/10.1371/journal.pcbi.1008752.

[28] M.G. Dodds, R. Krishna, A. Goncalves, C.R. Rayner, Model-informed drug repurposing: Viral kinetic modelling to prioritize rational drug combinations for COVID-19, Br. J. Clin. Pharmacol. (2020) 1–12, http://dx.doi.org/10.1111/bcp.14486.

[29] N. Néant, G. Lingas, Q. Le Hingrat, Jade Ghosn, Ilka Engelmann, Quentin Lepiller, A. Gaymard, V. Ferre, C. Hartard, J.-C. Plantier, V. Thibault, J. Marlet, B. Montes, K. Bouiller, F.-X. Lescure, J.-F. Timsit, E. Faure, J. Poissy, C. Chidiac, F. Raffi, A. Kimmoun, M. Etienne, J.-C. Richard, P. Tattevin, D. Garot, V. Le Moing, D. Bachelet, C. Tardivon, X. Duval, Y. Yazdanpanah, F. Mentré, C. Laouénan, B. Visseaux, J. Guedj, Modeling SARS-CoV-2 viral kinetics and association with mortality in hospitalized patients from the french COVID cohort, Proc. Natl. Acad. Sci. USA 18 (8) (2021) e2017962118, http://dx.doi.org/10.1073/pnas.2017962118.

[30] K. Ejima, K.S. Kim, S. Iwanami, Y. Fujita, M. Li, R.S. Zoh, K. Aihara, T. Miyazaki, T. Wakita, S. Iwami, Time variation in the probability of failing to detect a case of polymerase chain reaction testing for SARS-CoV-2 as estimated from a viral dynamics model, J. R. Soc. Interface 18 (2021) 20200947, http://dx.doi.org/10.1098/rsif.2020.0947.

[31] B.P. Holder, L.E. Liao, P. Simon, G. Boivin, C.A.A. Beauchemin, Design considerations in building in silico equivalents of common experimental influenza virus assays and the benefits of such an approach, Autoimmunity 44 (4) http://dx.doi.org/10.3109/08916934.2011.523267.

[32] B.R. Brückner, A. Janshoff, Importance of integrity of cell–cell junctions for the mechanics of confluent MDCK II cells, Sci. Rep. 8 (1) (2018) 14117, http://dx.doi.org/10.1038/s41598-018-32421-2, URL http://www.nature.com/articles/s41598-018-32421-2.

[33] C.A. Beauchemin, T. Miura, S. Iwami, Duration of SHIV production by infected cells is not exponentially distributed: Implications for estimates of infection parameters and antiviral efficacy, Sci. Rep. 7 (2017) 42765, http://dx.doi.org/10.1038/srep42765.

[34] Y. Kakizoe, S. Nakaoka, C.A. Beauchemin, S. Morita, H. Mori, T. Igarashi, K. Aihara, T. Miura, S. Iwami, A method to determine the duration of the eclipse phase for in vitro infection with a highly pathogenic SHIV strain, Sci. Rep. 5 (2015) 10371, http://dx.doi.org/10.1038/srep10371.

[35] H.M. Dobrovolny, C.A. Beauchemin, Modelling the emergence of influenza drug resistance: The roles of surface proteins, the immune response and antiviral mechanisms, PLoS One 12 (7) (2017) e0180582, http://dx.doi.org/10.1371/journal.pone.0180582.

[36] C.A. Beauchemin, J.J. McSharry, G.L. Drusano, J.T. Nguyen, G.T. Went, R.M. Ribeiro, A.S. Perelson, Modeling amantadine treatment of influenza a virus in vitro, J. Theoret. Biol. 254 (2008) 439–451, http://dx.doi.org/10.1016/j.jtbi.2008.05.031.

[37] L.T. Pinilla, B.P. Holder, Y. Abed, G. Boivin, C.A.A. Beauchemin, The h275y neuraminidase mutation of the pandemic A/H1N1 influenza virus lengthens the eclipse phase and reduces viral output of infected cells, potentially compromising fitness in ferrets, J. Virol. 86 (19) (2012) 10651–10660, http://dx.doi.org/10.1128/JVI.07244-11.

[38] B. Wendroff, Difference methods for initial-value problems (Robert D. Richtmyer and K. W. Morton), SIAM Rev. 10 (3) (1968) 381–383, http://dx.doi.org/10.1137/1010073, URL https://epubs.siam.org/doi/abs/10.1137/1010073.

[39] D.L. Olsen-Kettle, Numerical solution of partial differential equations 108.

[40] R. Bermejo, L. Saavedra, Lagrange-Galerkin methods for the incompressible Navier-Stokes equations: a review, Comm. Appl. Ind. Math. 7 (3) (2016) 23–52, http://dx.doi.org/10.1515/caim-2016-0021.

[41] P. Kim, D. Kim, Convergence and stability of a BSLM for advection-diffusion models with Dirichlet boundary conditions, Appl. Math. Comput. 366 (2020) 124744, http://dx.doi.org/10.1016/j.amc.2019.124744.

[42] Q. Xu, H. An, A class of domain decomposition based nonlinear explicit-implicit iteration algorithms for solving diffusion equations with discontinuous coefficient, J. Comput. Appl. Math. 386 (2021) 113232, http://dx.doi.org/10.1016/j.cam.2020.113232.

[43] S. Banei, K. Shanazari, On the convergence analysis and stability of the RBF-adaptive method for the forward–backward heat problem in 2D, Appl. Num. Meth. 159 (2021) 297–310, http://dx.doi.org/10.1016/j.apnum.2020.08.015.

[44] B.P. Holder, P. Simon, L.E. Liao, Y. Abed, X. Bouhy, C.A. Beauchemin, G. Boivin, Assessing the in vitro fitness of an oseltamivir-resistant seasonal A/H1N1 influenza strain using a mathematical model, PLoS One 6 (3) (2011) e14767, http://dx.doi.org/10.1371/journal.pone.0014767.

[45] E. Kosiachenko, Efficient GPU Parallelization of the Agent-Based Models Using MASS CUDA Library (Master of Science in Computer Science & Software Engineering), University of Washington, 2018.

[46] M. Lysenko, R.M. D'Souza, A framework for megascale agent based model simulations on graphics processing units, J. Artif. Soc. Soc. Simul. 11 (4) (2008) 10, URL https://www.jasss.org/11/4/10.html.

[47] P. Holvenstot, GPU-Accelerated Influenza Simulations for Operational Modeling (Master of Science in Computer Science), Western Michigan University, 2014.

[48] B. Shekh, E. de Doncker, D. Prieto, Hybrid multi-threaded simulation of agent-based pandemic modeling using multiple GPUs, in: 2015 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, 2015, pp. 1478–1485, http://dx.doi.org/10.1109/BIBM.2015.7359894.

[49] K. Blahut, C. Quirouette, J.J. Feld, S. Iwami, C.A.A. Beauchemin, Quantifying the relative contribution of free virus and cell-to-cell transmission routes to the propagation of hepatitis C virus infections in vitro using an agent-based model, arXiv arXiv:2102.05531.

[50] R. Laubenbacher, J.P. Sluka, J.A. Glazier, Using digital twins in viral infection, Science 371 (6534) (2021) 1105–1106, http://dx.doi.org/10.1126/science.abf3370.

[51] T. Li, Z. Cheng, L. Zhang, Developing a novel parameter estimation method for agent-based model in immune system simulation under the framework of history matching: A case study on influenza A virus infection, Intl. J. Mol. Sci. 18 (12) (2017) 2592, http://dx.doi.org/10.3390/ijms18122592.

[52] M.N. Read, K. Alden, L.M. Rose, J. Timmis, Automated multi-objective calibration of biological agent-based simulations, J. R. Soc. Interface 13 (122) (2016) 20160543, http://dx.doi.org/10.1098/rsif.2016.0543.

[53] L.J. Allen, E.J. Schwartz, Free-virus and cell-to-cell transmission in models of equine infectious anemia virus infection, Math. Biosci. 270 (2015) 237–248, http://dx.doi.org/10.1016/j.mbs.2015.04.001.

[54] N.L. Komarova, D. Wodarz, Virus dynamics in the presence of synaptic transmission, Math. Biosci. 242 (2) (2013) 161–171, http://dx.doi.org/10.1016/j.mbs.2013.01.003.

[55] S. Iwami, J.S. Takeuchi, S. Nakaoka, F. Mammano, F. Clavel, H. Inaba, T. Kobayashi, N. Misawa, K. Aihara, Y. Koyanagi, K. Sato, Cell-to-cell infection by HIV contributes over half of virus infection, ELife 4 (2015) e08150, http://dx.doi.org/10.7554/eLife.08150.

[56] F. Graw, D.N. Martin, A.S. Perelson, S.L. Uprichard, H. Dahari, Quantification of hepatitis c virus cell-to-cell spread using a stochastic modeling approach, J. Virol. 89 (13) (2015) 6551–6561, http://dx.doi.org/10.1128/JVI.00016-15.

[57] P. Kumberger, K. Durso-Cain, S.L. Uprichard, H. Dahari, F. Graw, Accounting for space - quantification of cell-to-cell transmission kinetics using virus dynamics models, Viruses 10 (4) (2018) 200, http://dx.doi.org/10.3390/v10040200.

[58] B. Jessie, H.M. Dobrovolny, The role of syncytia during viral infections, J. Theoret. Biol. 525 (2021) 110749, http://dx.doi.org/10.1016/j.jtbi.2021.110749.

[59] C. Quirouette, N.P. Younis, M.B. Reddy, C.A.A. Beauchemin, A mathematical model describing the localization and spread of influenza a virus infection within the human respiratory tract, PLoS Comput. Biol. 16 (4) (2020) e1007705, http://dx.doi.org/10.1371/journal.pcbi.1007705.
[60] G. González-Parra, H.M. Dobrovolny, The rate of viral transfer between upper and lower respiratory tracts determines RSV illness duration, J. Math. Biol. 79 (2019) 467–483, http://dx.doi.org/10.1007/s00285-019-01364-1.
[61] H. Manchanda, N. Seidel, A. Krumbholz, A. Sauerbrei, M. Schmidtke, R. Guthke, Within-host influenza dynamics: A small-scale mathematical modeling approach, Biosystems 118 (2014) 51–59, http://dx.doi.org/10.1016/j.biosystems.2014.02.004.
[62] J. Zheng, S. Perlman, Immune responses in influenza a virus and human coronavirus infections: an ongoing battle between the virus and host, Curr. Opin. Virol. 28 (2018) 43–52, http://dx.doi.org/10.1016/j.coviro.2017.11.002.
[63] H.M. Dobrovolny, M.B. Reddy, M.A. Kamal, C.R. Rayner, C.A. Beauchemin, Assessing mathematical models of influenza infections using features of the immune response, PLoS One 8 (2) (2013) e57088, http://dx.doi.org/10.1371/journal.pone.0057088.
[64] C. Kerepesi, T. Bakacs, T. Szabados, MiStImm: an agent-based simulation tool to study the self-nonself discrimination of the adaptive immune response, Theor. Biol. Math. Model. 16 (2019) 9, http://dx.doi.org/10.1186/s12976-019-0105-5.
[65] D. Levin, S. Forrest, S. Banerjee, C. Clay, J. Cannon, M. Moses, F. Koster, A spatial model of the efficiency of T cell search in the influenza-infected lung, J. Theoret. Biol. 398 (2016) 52–63, http://dx.doi.org/10.1016/j.jtbi.2016.02.022.
[66] H.M. Dobrovolny, M.J. Baron, R. Gieschke, B.E. Davies, N.L. Jumbe, C.A.A. Beauchemin, Exploring cell tropism as a possible contributor to influenza infection severity, PLoS One 5 (11) (2010) e13811, http://dx.doi.org/10.1371/journal.pone.0013811.
[67] A. Timm, J. Yin, Kinetics of virus production from single cells, Virology 424 (2012) 11–17, http://dx.doi.org/10.1016/j.virol.2011.12.005.
[68] R.G. Crystal, J.B. West, The Lung: Scientific Foundations, Raven Press Ltd., New York, USA, 1991.

**Baylor G. Fain** is a graduate student currently pursuing his Ph.D. in biophysics at Texas Christian University. His research interests are in applying physics to biological systems and parallization on desktop GPUs. He has explored this research in computer simulations of SARs-CoV-2 in-host infections and Influenza in-host infections.

**Hana M. Dobrovolny** received her B.Sc. from the University of Winnipeg in 1997 and a Ph.D in physics from Duke University in 2008. She is currently an Associate professor at Texas Christian University, where she has been studying viral dynamics since 2012.