1 Algorithm of the Computational Model

The main sections in the algorithm, shown in Fig.1 as a flow chart, consists of a statistical loop, infection propagation loop and a virus transmission loop. As the model outcome is stochastic, the average of many simulations with the same initial and boundary conditions yields results close to a deterministic model. Thus in the first loop, simulation is repeated several times for estimating the statistical parameters. All the model parameters such as infection characteristics and spread conditions are specified, and lattice computations occur in the second loop called the infection propagation loop.

The third loop, called the transmission loop, screens the transmittable neighbourhood and outputs the set $\{C^s_{ij}\}$. The range r and number R_t are specified in this loop. For the cell $x^t_{ij} = I$, the neighbouring cells are identified as a finite set $N_{ij} = \{(i + \alpha_k, j + \beta_k) : (\alpha_k, \beta_k) \in D\}$. Here D is a finite set of directional indices of cells in the transmittable neighbourhood. All the 48 neighbourhood cells are screened for examining its state, sequentially in the range r, but randomly in each layer, to eliminate any directional bias in propagation.

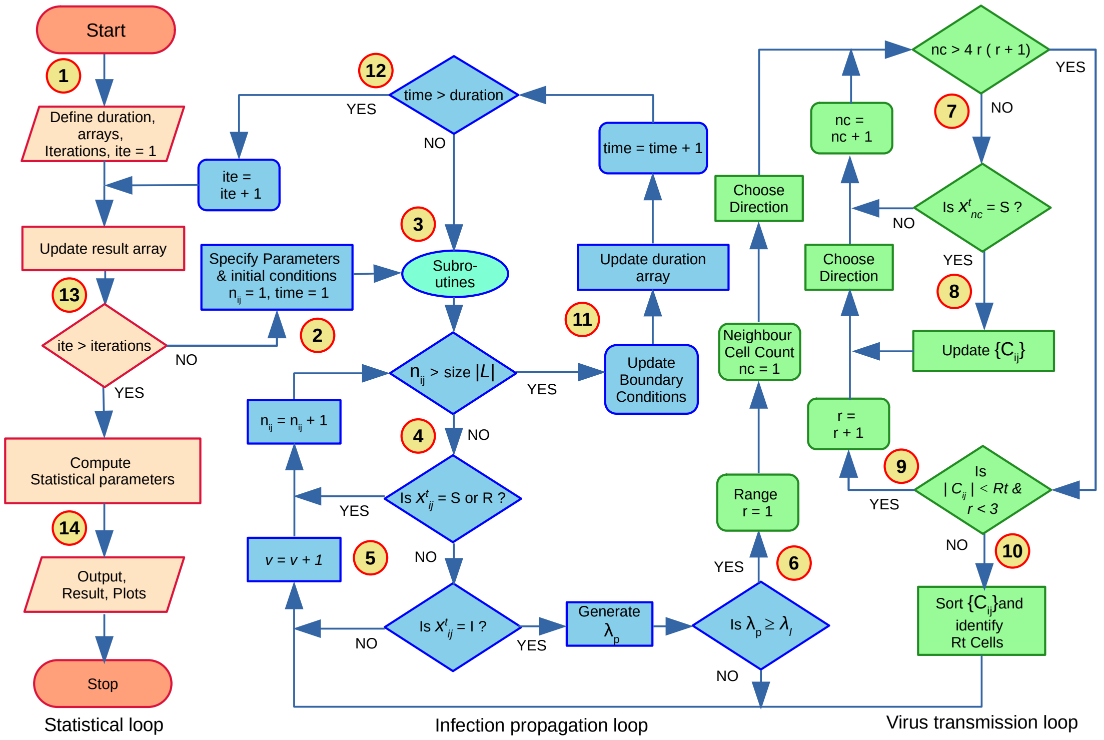


Figure 1: Flow chart of the programming algorithm

The important steps in the algorithm, as numbered in the flow chart are briefly described below.

1. The statistical loop is initiated and the duration, the number of iterations and array for storing results are defined.

2. In the infection propagation loop, the parameters and initial conditions are specified. This loop is restarted for each statistical iteration by resetting parameters and arrays.

3. Simulation is started with $time = 1$, and the relevant subroutines for lockdown, migration or vaccination, if specified, is called on reaching that particular day.

4. Each cell in the lattice $\mathcal{L}^t$ are randomly checked & updated starting with cell count $n_{ij} = 1$. State value of susceptible and recovered cells are not updated.

5. Cells not in infectious state and infectious cells which do not satisfy the condition $\lambda_p \geq \lambda_l$ are updated by $v = v + 1$.

6. Cells in the infectious period, satisfying the condition $\lambda_p \geq \lambda_l$, are passed to the virus transmission loop.

7. In the virus transmission loop, the transmittable neighbourhood of infectious cell is scanned to identify potential S state cells. The screening is started with range $r = 1$ in a random direction.

8. The details of cells in the susceptible state are stored in an array as $\{C_{ij}\}$.

9. If R_t number of S state cells are not obtained, the next range of neighbourhood cells are scanned.

10. The identified cells are sorted according to their sum of state values and R_t cells with the maximum sum of state values ($\{C^s_{ij}\}$) are marked as infected, by updating $v = 1$.

11. After checking all cells in the lattice, the boundary conditions are updated. The results from the evolution of the lattice for the time step are stored in a duration array.

12. If time $<$ duration, do simulation for next time step, to find future evolution of states in the lattice.

13. If time $>$ duration, the results array in the statistical loop is updated by copying duration array to it and the next iteration is started.

14. After completion of specified iterations, the statistical parameters are calculated and the results are generated.

1.1 Migration subroutine

1. Create a new lattice of dimension higher than m by an incremental value δ.

2. Map the states of the existing lattice to the new lattice.

3. Randomly mark the migrating exposed states in the new lattice.

1.2 Vaccination subroutine

1. Input the variables d_v, C_v and r_v.

2. Compute R_v, the per-day registration for vaccination.

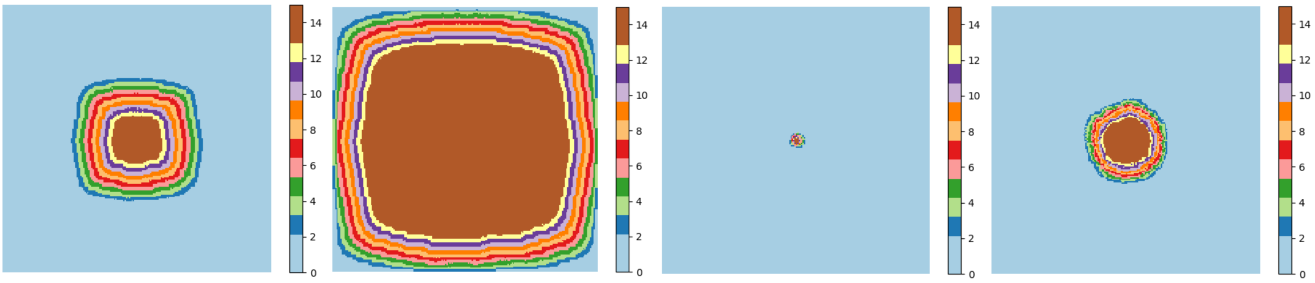
3. Randomly select cells in susceptible state and update to vaccinated state until the number of per day vaccinated is equal to per day registration.

Algorithm 1 Pseudocode for Virus transmission loop

```
Input the lattice states and indices of infected cell, range r=1
while Range $r \leq 3$ do
    Generate the directional indices of transmittable neighbourhood for the infected cell
    for Each cell in the transmittable neighbourhood do
        if Cell is not a boundary cell and is in susceptible state then
            Generate the directional indices of it's transmission neighnourhood
            Calculate the sum of state values of transmission neighbourhood
            Store the sum and indices of cell as list of tuples = Result
        else
            Continue to next cell
        end if
    end for
    if Result have $R_t$ cells to infect then
        return  Result list
    else
        Move to next range r=r+1
    end if
end while
return  Result
```

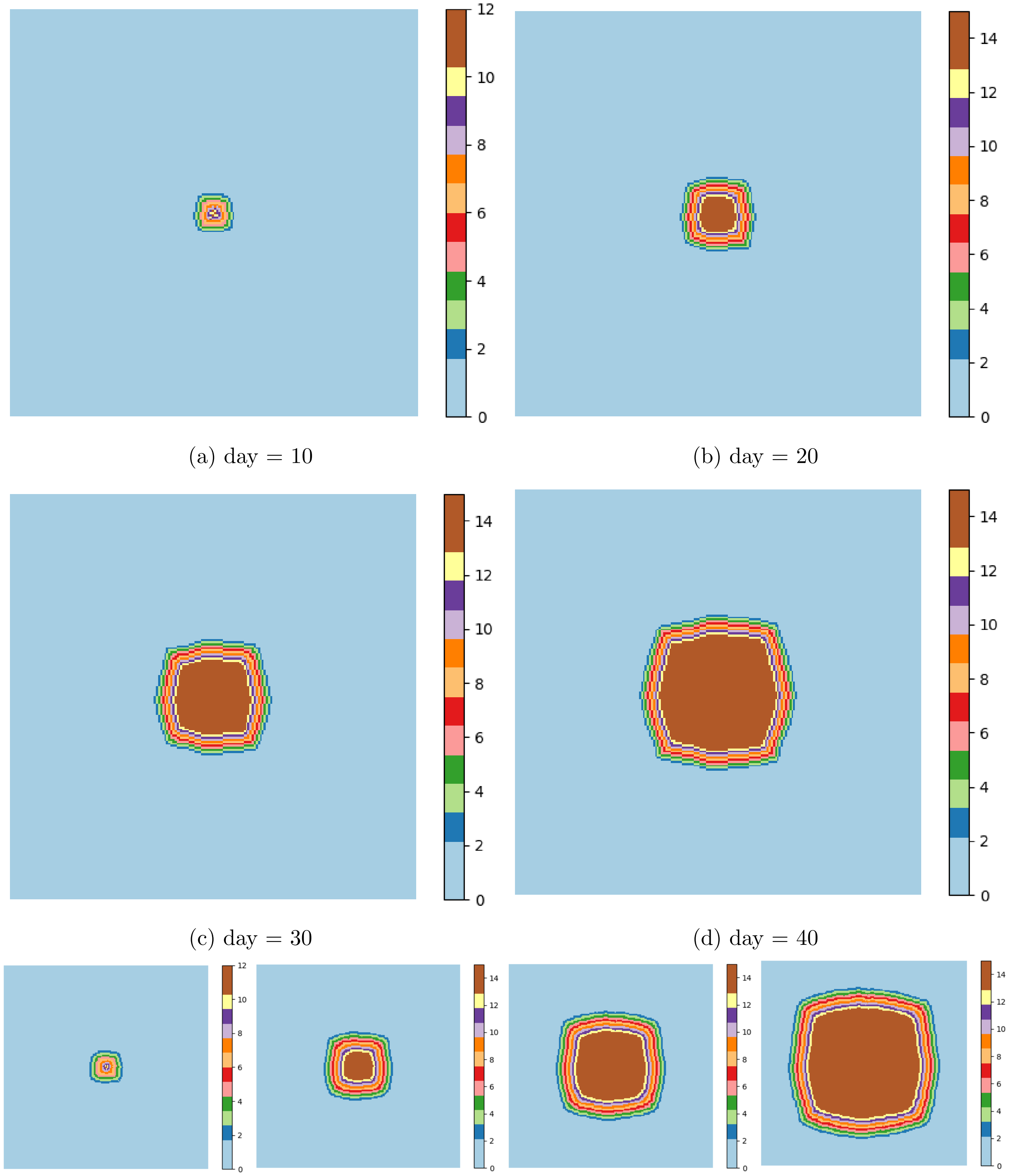
2 Contour plots of the evolution of CA

Temporal evolution of the lattice cells during pandemic propagation is shown as colored states for better data visualization in this section. Simulating restricted interaction during lockdown is shown in Fig.2. Figure 3 indicates that as range of the transmittable neighbourhood is increased, infection propagates to more people for same time period. Similarly, the extent of propagation also depends on initial placement of infected cell in the lattice as shown in Fig.4. The evolution of the lattice with multiple infected/immune cells randomly distributed initially, is shown in Fig.5. The color bar and numerical values represents the state values v of each cell.



(a) day = 20 (b) day = 40 (c) day = 20 (d) day = 40

Figure 2: Effect of social isolation factor for locality on propagation. For figures a-b $\lambda_l = 0$ and for figures c-d $\lambda_l = 0.75$ from day 1. Parameters $m = 200, I = 1$.



(a) day = 10

(b) day = 20

(c) day = 30

(d) day = 40

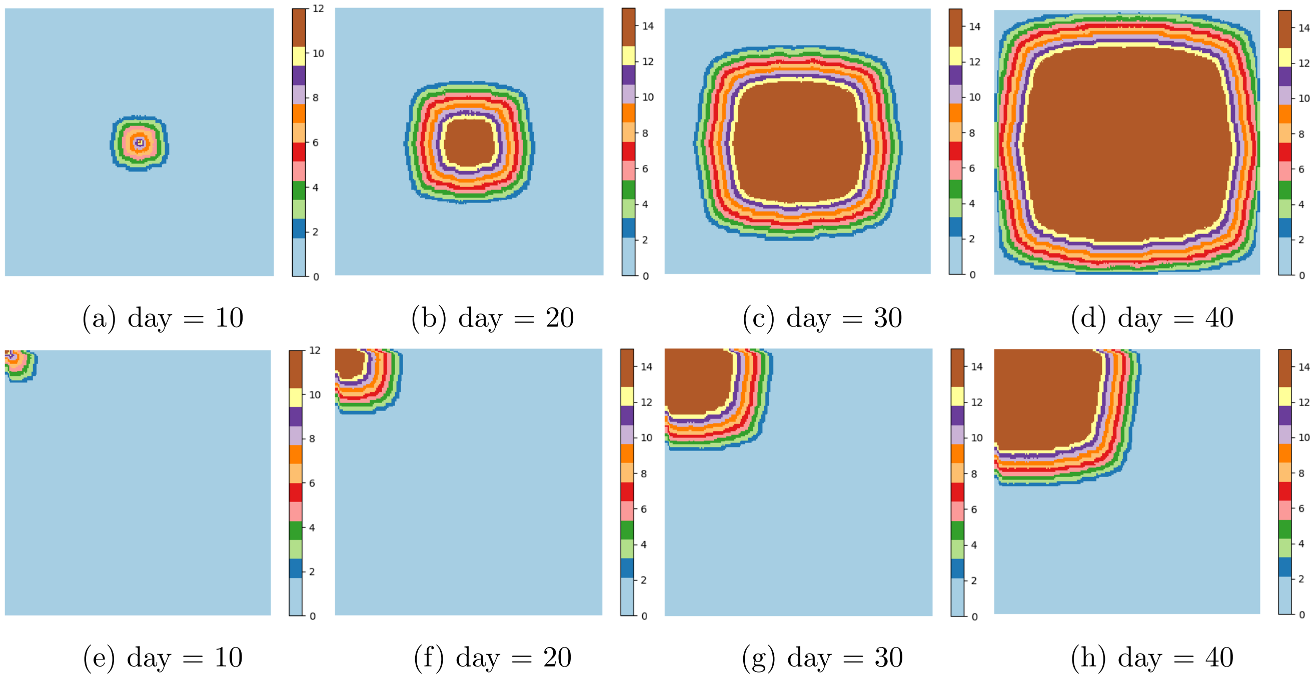
(e) day = 10

(f) day = 20

(g) day = 30

(h) day = 40

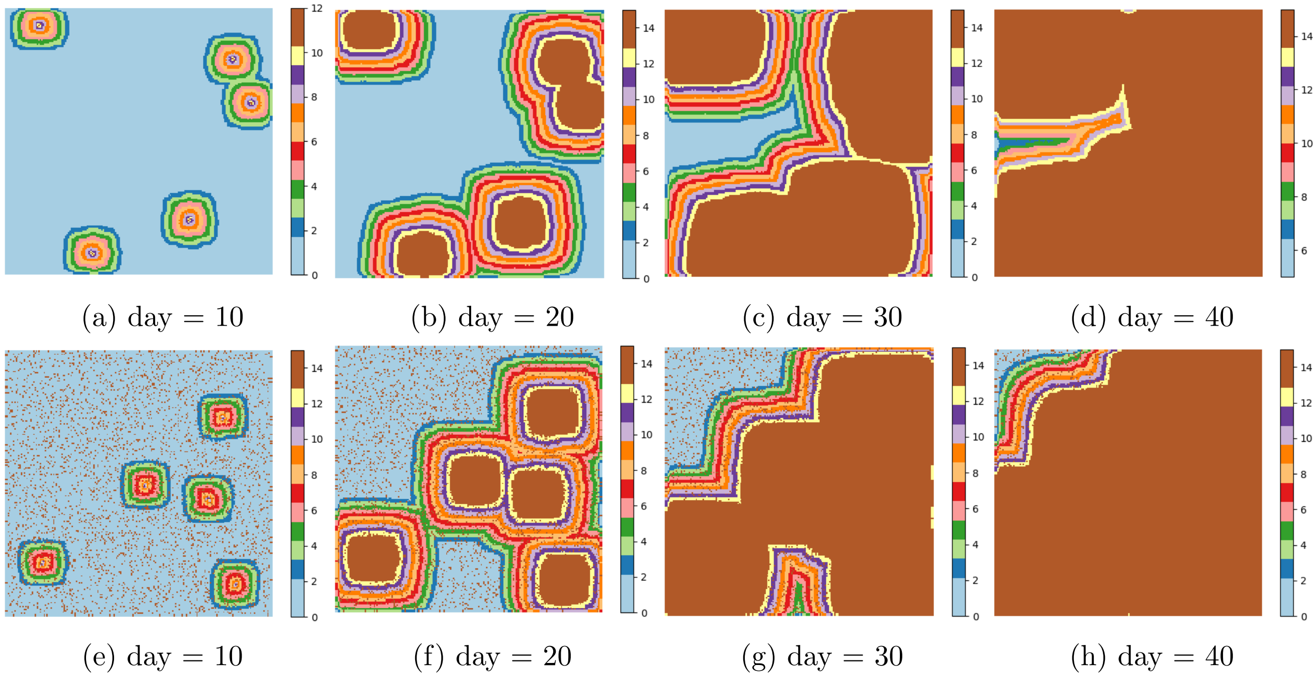
Figure 3: Effect of range of transmittable neighbourhood on temporal evolution of the cellular automata. For figures a-d, range $r = 1$ and for figures e-h range $r = 2$. Parameters $m = 200, I = 1, \lambda_l = 0$



(a) day = 10 (b) day = 20 (c) day = 30 (d) day = 40

(e) day = 10 (f) day = 20 (g) day = 30 (h) day = 40

Figure 4: Effect of initial placement of infected cell in the lattice on evolution of the cellular automata. For figures a-d, the infected cell is initially at the center of lattice, and for figures e-h, the infected cell is initially at a corner of the lattice. Parameters $m = 200, I = 1, \lambda_l = 0$ and range $r = 3$.



(a) day = 10 (b) day = 20 (c) day = 30 (d) day = 40

(e) day = 10 (f) day = 20 (g) day = 30 (h) day = 40

Figure 5: Evolution of the cellular automata with and without immune population. Figures a-d without initial immune population and figures e-h with 5% immune population at day=0. Initial infected cases are randomly placed in the lattice. Parameters $m = 200, r = 3, I = 5, \lambda_l = 0$.

3 COVID-19 propagation in an age group

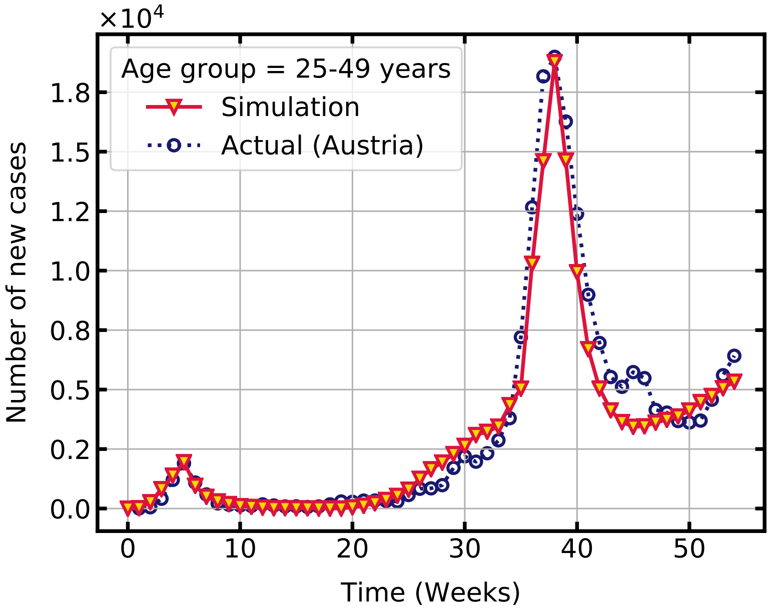


Figure 6: Comparison of simulated and actual weekly new cases of COVID-19 in a particular age group. Data source: European Center for Disease Prevention and Control (https://www.ecdc.europa.eu/en accessed on 14 August 2021)

Table 1: Model parameters used in simulation of Fig.6

Common parameters (Notation)	Value
Initial parameters	
Lattice dimension (m)	100
Infection parameters	
Latent period (L_p)	2 days
Infectious period (I_p)	3 - 6 days
Recovery time (t_R)	14 days
Social isolation factor (λ_l)	0
Change in parameters	
Day 35	$\lambda_l = 0.85$, $\delta = 0$
Day 130	$\lambda_l = 0.70$, $\delta = 1$
Day 245	$\lambda_l = 0.40$, $\delta = 4$
Day 267	$\lambda_l = 0.75$, $\delta = 1$
Day 285	$\lambda_l = 0.79$, $\delta = 1$